This dissertation has been assessed by:

. . . .

DISSERTATION

The Space of People in Architectural Computation

a conceptual framework for the design computation of human-centric spatial environments

conducted for the attainment of the academic degree of
doctorate of technical sciences supervised by

Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Christian Kühn

E253
Institute of Architecture and Design

Submitted at the Vienna University of Technology
Faculty of Architecture and Space Planning

by

Dipl.-Arch. Christian Wilhelm Derix

student number (Matrikel) 0928529
Karlsplatz 13, 1040 Vienna

Vienna, 17.12.2015

advised by

Emeritus Univ.Prof. Dipl.-Arch. Dr.techn. Philip Steadman

Principal Research Associate
Bartlett School Environment, Energy & Resources
Faculty of the Built Environment
University College London

"The essence of architectural form is infected with movement, 'for through the experiences of our visual sense [...] come to rest the intuition forms of the three-dimensional spaces'." (Schmarsow 1894, p15; in Schwarzer 1991, p59)

ABSTRACT

Ever since its mainstream introduction to architectural design, computation has been treated as an engineering tool, forcing architects into problem-solving workflows for efficient containers. The core focus of architecture to develop environments for human agency beyond basic needs appears lost and to many impossible when computational representations are engaged. This impression is reinforced by much publicised developments in architectural computation that focus exclusively on design management and geometric shape exploration.

Paradoxically, since the introduction of computation in architecture in the 90s, the architectural profession has abandoned the modernist approach that provided the analogy of environments as efficient machines and reoriented itself towards the value of human performances as drivers of spatial environments. For computation to enable this reappearing design paradigm, new models need creating that represent architecture as a human-centric field rather than lifeless container.

Beyond R&D for the engineering paradigm of computation in design such as computer-aided design (CAD), building information modelling (BIM) and parametric modelling, three strands of architectural computation were explored by academics since the 1960s covering fundamental aspects of human-centric architectural design: a) self-organizing algorithms to generate spatial organization, b) algorithmic representations of space and c) analytical representations of spatial configurations and human performances. While aware of each other, the strands have remained largely isolated. A) investigated new *designing* methodologies for architecture through self-organizing algorithms, b) new *representations* of space through computational geometry and c) correlations between spatial *configuration* and *occupation*. What has been missing is their synthesis to develop a computational methodology for designing spatial environments based on human-centric performances.

The dissertation discusses the three strands of computational research into architectural design and demonstrates through case studies how the original academic models of those research strands can be translated to live design situations in practice. The synthesis of the three approaches will be demonstrated across two stages: firstly through direct integration and secondly through abstraction into a meta-system. The meta-system represents a conceptual computational framework for human-centric spatial design, which will be called user-centric spatial operations model (USOM). The currently active framework of applications called Open Framework for Spatial Simulation (OFSS) has been developed by the author with multiple colleagues through academic research and professional developments since 1999, representing the first concise framework of its kind and remains in continuous professional and academic use and development.

"Das Wesen der architektonischen Form ist durchzogen von Bewegung, so 'daß durch die Erfahrungen unseres Gesichtssinnes [...] die Anschauungsform des dreidimensionalen Raumes zu Stande kommt'." (Schmarsow 1894, p15; in Schwarzer 1991, p59)

ABSTRAKT

Seit ihrer Einführung in den architektonischen Entwurf wurden Computerberechnungen (Computation) als ein Ingenieurswerkzeug behandelt, der Architekten in einen Arbeitsablauf von Problemlösungen für effiziente Hüllen gezwungen hat. Wo computerbasierte Darstellung eingebracht wird, scheint der Kern der Architektur Umgebungen fuer menschliche Handlungen zu schaffen, die über elementaren Nutzen hinausgehen, meist verloren. Dieser Eindruck wird durch Veröffentlichungen bekannter Entwicklungen der computerbasierten Architektur unterstützt, die sich fast ausschliesslich dem Prozessmanagement und Formstudien widmen.

Im Gegensatz dazu, rückte das Berufsfeld der Architektur seit der Einführung architektonischen Computation in den 90er Jahren von dem modernistischen Ansatz ab, der die Analogie maschienenähnliche Umwelten postuliert hatte, und hat sich am Wert menschlichen Raumverhaltens als Leistungsprinzip für räumliche Umgebungen neu orientiert. Damit Computation dieses wiederkehrende Entwurfsparadigma befähigen kann, müssen neue Modelle entwickelt werden, die Architektur als benutzer-orientierte Umgebung statt leblose Hülle darstellen.

Jenseits der effizienzbedingten Entwicklungen für computerbasiertes Entwerfen , wie computer-aided design (CAD), building information modelling (BIM) und parametrisches Modellieren wurden drei architektonische Ansätze des Computation durch Akademiker seit den 60ern erforscht, die grundsätzliche Aspekte des benutzer-orientierten Entwurfs abdeckten: a) selbst-organisierende Algorithmen zur Generierung räumlicher Organisationen, b) algorithmische Darstellung von Raum und c) analytische Darstellung von räumlichen Gestalten und menschlichen Raumverhaltens. Obwohl die drei Richtungen sich berührten, blieben sie klar voneinander getrennt. A) erkundete neue architektonische *Entwurfsmethoden* durch selbst-organisierende Algorithmen, b) neue *Raumdarstellungen* durch algorithmische Geometrie und c) Wechselwirkungen von räumlicher *Gestalt* und Benutzung. Ausgeblieben ist ihre Synthese zur Entwicklung einer computerbasierten algorithmischen Methodik für die Gestaltung räumlicher Umgebungen fundiert auf benutzer-orientierten Leistungen.

Die Dissertation erläutert die drei Bereiche der computationalen Forschung für den architektonischen Entwurf und zeigt anhand von Fallbeispielen wie deren ursprünglichen akademischen Modelle in Projektentwurfsprozessen der Praxis angewandt werden können. Die Synthese der drei Ansätze wird über zwei Abschnitte diskutiert: erstens durch unmittelbare Verbindungen und zweitens durch eine Abstraktion in ein Meta-System. Das Meta-System stellt ein offenes rechenbasiertes Rahmenkonzept für benutzer-orientierte räumlichen Entwurf dar, das 'benutzer-orientiertes Raumverfahrensmodel' gennant wird (USOM: user-centric spatial operations model). Die Methodik und das daraus resultierende derzeitig aktive

Anwendungsframework – Open Framework for Spatial Simulation (OFSS) genannt – wurden von dem Author mit Kollegen in akademischer Forschung und professioneller Entwicklung seit 1999 erstellt, und stellt das erste pägnant zusammengeführte Anwendungsframework seiner Art dar, das durchweg in professionellem und akademischen Gebrauch und Entwicklung ist.

INDEX

ACKNOWLEDGEMENTS

I would like to thank Professor Manfred Wolff-Plottegg for encouraging me to start this journey seven years ago. And I am very grateful to my supervisors Professor Christian Kühn and Professor Philip Steadman for their clear guidance, technical support and patience during the production of this dissertation. Additional support was provided by Professor Kristina Shea during the early structuring phase. Generally, I would like to express my appreciation for the Faculty for Architecture and Planning at the Technical University Vienna, which provided a stimulating environment for ten years.

The core work and research thrust would not have happened without the late Paul Coates at the University of East London, who was my teacher, mentor and good friend. Countless insightful discussions and many projects with Paul have laid the foundations for this dissertation. He also connected me with the late Alasdair Turner at the University College London who also shared many inspiring conversations and technical skills. I would also like to thank Robert Thum who pointed me towards self-organizing neural networks during my masters programme. In the field of connectionism and urban planning Professor Lidia Diappi of the Politecnico di Milano offered novel arguments and continuous support. For many years the chair for Information Architecture at the ETH Zurich invited me to discuss my work and provided thus valuable practice to rehearse the arguments.

After the first years of self-driven research and development in academia, I have been privileged to work with a highly skilled and knowledgeable group of people in practice-based research: Pablo Miranda Carranza, Åsmund Izaki, Lucy Helme, Prarthana Jagannath and Anders Holden Deleuran. Without this team, the dissertation would not have been possible in its current format, because as a team we developed many of the case study projects discussed.

I would also like the thank Peter Oborn who as director of the Aedas London office supported my endeavour to do a PhD by creating a context within which the academic and the professional work could coexist. Ross Donaldson as managing director of Woods Bagot facilitated a similar role, which allowed the conclusion of this work.

Finally, my wife Eleanor was an incessant motivator and supporter who managed the family with enduring strength over years, cities and babies. I owe this work to her.

1 FROM CONTAINERS TO FIELDS

"Space between things turns out not to look simply empty." (Arnheim 1974, p17)

1.1 MOTIVATION

The emphasis on the correlation of space-to-occupant in architecture fluctuates across architectural design paradigms. Since the onset of the industrial revolution the focus of architectural design seems to have firmly shifted towards its modes of production as Robin Evans also suggested: *"the social aspect of architecture, […], was more concerned with the fabrication of buildings than with their occupation. [..] Emphasis shifted from the nature of the place to efficient procedures of its assembly."* (Evans 1997, p11). The introduction of computer-aided design (CAD) to mainstream architectural practice since the early 1990s has exacerbated this situation. More recent applications of computation in the profession have cemented this approach, enabling architectural designers to control visible geometries and their construction (parametric modelling), share and access component specifications (building information modelling – BIM). Where users are addressed, aspects of occupation are mainly attributed to explicitly quantifiable yet indirect performances such as climate (comfort) or logistics (operations, scheduling or egress). When looking at computational tools and performances employed to design architecture today, the impression arises of architecture as a lifeless human-free object. Or vice-versa, of computation as a limited engineering tool.

Architectural education in the 1990s in the UK involved a mixture of modernist and post-modernist thinking with modernist objective rules for rational spaces and post-modernist subjective intuition for phenomenological spaces. Students would read post-modernist theories relating to social and cognitive themes while reviewing modernist buildings (including 'high-tech'). Design teaching at the *atelier* schools in the UK[1] focussed on post-modernist design theories, emphasizing subjective mappings to conjure bespoke architectural organizations of space by induction, which Colin Rowe called *maximum non-interventionism* (Rowe 1983, p12). Maximum non-interventionism represented a blend of modernist and post-modernist design methodologies by first engaging in rigorous quasi-scientific analysis of a context and secondly arriving at a highly personalized human-centric interpretation into spatial organization. While computation in architectural design was taught primarily as CAD support, Paul Coates at the University of East London (UEL) and John Frazer at the Architectural Association taught algorithmic thinking and some computer programming. Their aim was however to avoid analogue architectural design altogether in favour of a *new epistemology* through cybernetic systems encoded using self-organizing algorithms (Coates 2010). Coates and Frazer's cybernetic approach fitted the scientific mapping of 'maximum non-interventionism' well but did

[1] In 1990s principally: the Bartlett of the University College London, the Architectural Association, the University of East London and the University of North London

not agree with subjective design interpretations for spatial organization. This seemed paradoxical, because particularly Coates' self-organizing algorithms for distributed systems of space represented situated users of spatial environments. As a student of Paul Coates and educated in both modernist and post-modernist theory and design methodology, the author started to question the enforced lack of correlation between computational cybernetic systems and the post-modernist emphasis on subjective perception of space such as described through Pallasmaa's phenomenology (Holl et al 1994). The initial instinctive research question therefore already emerged during the hybrid UEL programme of the diploma design studio with MSc Computing & Design in 1999 and envisaged the investigation of the relation between algorithms and cognition, starting with the development of the first self-organizing artificial neural networks for architectural representation (Derix and Thum 2000; Derix 2001).

All ingredients appeared available to envisage an outline brief for a computational framework to develop a design methodology that embeds correlations between spatial organization and human-centric performances, such as:

- *generative self-organizing algorithms* based on distributed representation by mainly Coates (2010) and Frazer (1995) constituted an epistemological equivalent to modernist cybernetics and post-modernist context-based generative systems. Their use of meta-heuristic algorithms correlated well to heuristic design methodologies of architects like Hans Scharoun (Janofske 1984) or Herman Hertzberger (2005) who drove rule-based aggregations from human-centric conditions; and contextual (de-)composition systems like Peter Eisenman's (1963) or Bernard Tschumi's (1994) syntactical fields
- representations of *spatial cognition* for the study of human perception of space such as proposed by James J Gibson (1950), Kevin Lynch (1960) or Michael Benedikt (1979); those studies found translations into architectural research seeking configurational representations of spatial environments that correlate to user occupation as developed by Bill Hillier and colleagues at University College London (UCL) (Hillier and Hanson 1984)
- *parametric representations of spatial structures* as efficient computational systems by researchers such as Philip Steadman (March and Steadman 1971) or Philip Tabor (1971) at the Centre for Land-Use and Built Form Studies (LUBFS) at Cambridge University. These mathematical representations provided important computational foundations for professional design implementations of generative and configurational concepts such as graphs and networks
- design workflow and knowledge representations by the knowledge-based design community (KbD) through researchers such as Charles Eastman (1969), John Gero (1990) or Richard Coyne (1988), providing pilot *ontologies of the design process* and domains into computational data structures
- *system theory* of various disciplines having explored different representations of complex systems such as the design of spatial environments by Herbert Simon (Simon 1969), Christopher Alexander (1964), Heinz von Foerster (1984), Humberto Maturana (1978), Hermann Haken (Haken and Graham 1971) or Bruno Latour (1987)

These strands of research were not fully isolated yet never synthesized into a design methodology through computation. In 2006, the author presented a paper that anticipated the research objectives of this dissertation (Derix, 2006) outlining the need for computation to be situated more in architectural thinking than scientific modelling. In hindsight, the proposal that architectural phenomena could be approximated via computational epistemology represents a continuation of the approach of the 19th century aesthetics and empathy theories. Those were first discussed in architecture by August Schmarsow (1894) and later translated into an analogue design methodology by early organic architects like Haering and Scharoun (Joedicke 1982). Essentially, the dissertation provides a prototypical implementation of the early 20th century empathic and organic methodology into computational design, synthesizing the above listed design research strands of the second half of the 20th century.

1.2 DISAMBIGUATION

As many of the terms used in the title and introduction of the dissertation are ambiguous within architectural theory a short clarification of some basic concepts is provided:

1.2.1 Computational Design

In 2015, Computational Design has become a broad term for the use of computers in architectural design, mainly referring to what used to be called Parametric Modelling during the 2000s. When the author set up the R&D group at Aedas architects in London in 2004, two internal strands were carefully distinguished: Advanced Modelling for parametric modelling through commercial software like Bentley's Generative Components[2] and Computational Design for spatial design using algorithmic systems. The term was borrowed from the 1960s field of Computer Art (Kluetsch 2007) and constituted therefore the first professional Computational Design group in architecture, intended to distinguish spatial design computing from geometric shape rationalization. This dissertation maintains this distinction and intends computational design to represent spatial design simulation through computational algorithms.

Apart from very few exceptions, computational design in academia still adheres to the principles of parametric modelling, concentrating on form-finding and shape rationalization. This interpretation of computation in architecture is expressed through single-space pavilions littering every university campus featuring a computational design department.

1.2.2 Space and Occupation-based Computational Design

Instead of defining the role of the occupant as user, the dissertation defines properties adhering to occupants and other spatial users as *human-centric* instead of *user-centric*. The term user-centric has associations in various creative industries

[2] http://www.bentley.com/en-US/Promo/Generative%20Components/default.htm, accessed 03.01.2015

such as users of soft- and hardware in information technologies. In architectural design, the notion of the user also evokes profiles of functionally premeditated activities such as user profiles in workplace design (Vischer 2007). This type of functional user provides also the basis for contemporary post-occupancy evaluations (POE) partially conducted through computational means. This dissertation intends the user to be more generally conceived as an occupant in spatial environments and therefore his agency not being restricted to function-specific activities but open to general behaviours and cognitive facilities. The user will therefore also be discussed as designer who uses design behaviours (heuristics) and software. This distinction is further explained in 1.2.4.

Human behaviour and cognition in the context of spatial environments are always related to geometric representation. The discipline of spatial cognition analyses the user as occupant based on many other non-geometric properties such as visual signal processing, memory or semantic networks (Montello 2001), which are not touched on. The basis for the analysis of human-centric affordances of space in this dissertation refers to the architectural abstraction into geometric drawings and models representing boundary properties of spatial environments. Hence, human-centric performances always correlate to the configuration of spatial elements. The *field* as used in this dissertation represents the space of conditions that result from the mapping of human users and geometries of spatial configurations. This principle is borrowed from Bill Hillier's original space syntax theory: *"[…] the minimum synactical rule for a spatially coherent aggregate, and its global result is that, through the distributed repetition of its local rule it defines the carrier space."* (Hillier et al, 1976). But the term *field* itself is borrowed from Stan Allen's essay From Object to Field (Allen 1997).

The correlation between space and occupant as human user represented via computational algorithms revolves mainly around the work of two British researchers: Bill Hillier and Paul Coates. While Bill Hillier and colleagues at UCL developed theories for the design of spatial configurations via syntactical rule sets, it was Paul Coates who introduced Hillier to cybernetics and computational syntaxes, namely algorithms. Coates encoded Hillier's original Space Syntax (Hillier et al., 1976) into generative algorithms in 1979 which were published in The Social Logic of Space (Hillier and Hanson 1984). Hillier continued to focus on human-centric spatial analysis while Coates continued onto self-organizing algorithmic system for the generation of spatial organizations. Apart from Coates' translation of Hillier's syntaxes of 1976 into algorithms, no other attempt existed for a concerted effort to synthesize those two fields. The research objective of this dissertation in the simplest sense would be the *synthesis of human-centric spatial analysis with self-organizing generative algorithms*. The two main works of those researchers provide the core foundation for the dissertation: Hillier's Space is the Machine (1996) and Coates' Programming Architecture (2010).

1.2.3 <u>Discursive Aspects of Human Properties in Design and Terminology</u>

Throughout the dissertation discursive non-scientific terminology will be used such as 'experience' or 'experiential', 'association', 'phenomena', 'empathy' or 'heuristic'. Due to some developments in architectural research those terms might provoke unintentional discomfort to some readers. To avoid confusion, some terms are briefly explained:

- Experience/experiential: empirical knowledge
- Association: relating two conditions, processes or data types; in parametric modelling, associations are used to linearly link code statements; this dissertation does not exclude this causal link but also intends cognitive association
- Phenomena: spatial occurrence from associating two processes, qualities or quantities such as described by the Gestalt theorists (Arnheim 1974)
- Empathy: an isomorphic correlation between objects and subjects as described by the late 19th century German aestheticists (see 1.4.3)
- Heuristics: rules-of-thumb or learned procedures from repetition to attain a satisfactory goal state (Simon and Newell 1958); this applies also to computational heuristics where a known one-off code helps resolve contextual issues (similar but different from *meta-heuristics*, see chapter 3)
- Affordance: properties of space that provide potentials for action when perceived by the observer; coined and elaborated by James Gibson (1979). Affordances also apply to implicit configurational properties that Hillier (1996, p31) called *non-discursive* properties of space encoding social and cultural norms. However, also spatial configurations need to be experienced visually and through movement as argued by Schmarsow (1894) or Gibson (1950)

Other discursive terminology used in the dissertation will be explained upon occurrence. The use of 'creativity', 'instinct' or 'intuition' in relation to some presumed design genius is omitted or will be explained where occurring. In knowledge-based design (KbD) theory, those anthropologic design qualities have been mostly unsubstantiatedly claimed to explain unquantifiable decisions (Gero and Maher 1993).

1.2.4 Behavioural Assumptions and Algorithmic Correlation

A series of case studies use assumptions about behavioural performances of virtual occupants in space. Insights about behavioural or perceptual affordances are correlated to algorithmic representations and procedures that have neither been explicitly researched nor scientifically validated by the author but draw on research by other academics, whose work is not questioned here. Some validation has taken place by trialling representations with sector experts who could confirm and propose assumptions about correlations from their professional experience. Additionally, some correlational validation has taken place occasionally on projects via university courses (such as the agent-based work) or through professional evaluation (such as pedestrian mapping). It is nonetheless a clear shortcoming in practice (and often academia) that observational validation, i.e. some scientific rigour, is not more often applied to project work.

Further, most algorithmic models do not originate from architectural research but borrow from other disciplines where spatial performances or social behaviours have been investigated. Particularly, biological and robotic concepts are treated specifically as *analogies* and never claim to be literal representations of the end-user of architectural space. Mostly, those analogies provide processing meta-heuristics that help to approximate design process heuristics or in small part also some insights for user behavioural heuristics.

Academic research for spatial behaviours this dissertation heavily draws on includes amongst others:

- correlations between geometric configuration and occupation (Lynch 1960; Hillier et al 1976; Hillier 1996, Turner et al 2001; Franz and Wiener 2008)
- graph theory and network analysis (Dijkstra 1959 ;March and Steadman 1971; Freeman 1977; Hillier and Hanson 1984; Wasserman and Faust 1994; Turner 2000)
- movement patterns (Reynolds 1987; Conroy-Dalton 2003; Arthur and Passini 2002)
- visual performances (Gibson 1979; Benedikt 1979; O'Rourke 1994; Turner et al 2001)
- cognitive classifications (Teuvo Kohonen 1995)

Additionally, the above mentioned academic research uses statistically and geometrically normalized behaviours and thus excludes subjective actions that deviate from norms. As described in section 6.4.2 this reduction is intentional since individual behaviours are firstly more difficult to approximate because they depend on subjective perceptions, cultural and social conditioning as well as preferences towards variations of dynamic contextual conditions (like daily weather patterns); and secondly, in a project design setting (the dissertation aims at *designing* situations rather than spatial analysis) a building brief for large complex spatial environments is seeking to accommodate 'populations' of users rather than individuals. It can also be assumed that deviations in individual behaviours will converge over time towards some behavioural pattern. A building or place is meant to provide inclusive utility to all its users without exclusion due to over-specification toward some outliers. Having said that, it is increasingly possible to specify individual user behaviours via new digital sensory monitoring methods but during the compilation of the dissertation this technology was not wide-spread and its effects are yet unknown, i.e. results of individualistic mappings have not been validated to perform better than population-based mappings (Derix and Conroy-Dalton, 2015).

However, it is specifically one of the author's believes that algorithmic design can be applied to subjective sensations and the design of spaces that afford individual correlations and this approach has been outlined in 8.2 Reflective Search. The key difference to purely academic analysis of individual behaviours was the specification of a design workflow that can accommodate subjective sensations translated into associated algorithmic models. Other projects of this nature have been omitted due to the focus of the dissertation on generalizable models of space-user correlations.

The dissertation or the case study models discussed don't claim to produce predictions of real conditions or simulations of real-world behaviours but approximations of behavioural patterns that help to weight generative processes.

1.2.5 Observer and User

One of Coates' favourite concerns was the status of the observer, incessantly pointing out that in programming the global observer and local actor are mediated via the code structure: "*person looking at the computer using brain/eye looking at global observer in the program – computer observer of - local agents in the program who just observe their immediate environment*" (Coates 2010, p47).

A common perception in Research and Technological Development casts the 'user' as the user of software, which Coates above described as 'person'. Here the term user usually refers to the citizen/occupier of a spatial environment (see 1.2.2). With 'observer' mainly the designer is intended who executes user-agencies such as observer of occupant behaviours that he simulates mentally (empathy) while designing, as observer of computational processes visualized before him that attempt to communicate certain associations. The specific type of observer is usually suffixed with 'observer-designer' or 'observer-user'.

Hillier and Leaman paved the way for Hillier's concept of the *inverse law* (Hillier et al 1976) by contemplating the dual relationship between the heuristics of use and heuristics of the observer as designer. The *inverse law* – or *manifold structure* as they first called it (Hillier and Leaman 1974), specified a correlation between acting in the environment due to learned behaviours isomorphic to spatial features. Design heuristics essentially encode the generation of those spatial features that provide the environmental agencies to users of space according to conventional learned actions. Two heuristics are mutually co-evolving.

For brevity, the dissertation has reduced the gender of the user/observer to its masculine form.

1.3 RESEARCH OBJECTIVES

The key research objective for this dissertation has been explained in the introduction and in section 1.2.2 and is summarized to:

OBJECTIVE 1: CORRELATE HUMAN-CENTRIC PERFORMANCES TO ASPECTS OF SPATIAL CONFIGURATION THROUGH GENERATIVE COMPUTATIONAL SYSTEMS.

Hillier and colleagues' original Space Syntax theory constitutes *a theory of morphic languages* encoding generative syntaxes for social patterns of space (Hillier et al. 1976, p149). This led Paul Coates to encode the generative syntax because he was himself interested in distributed representations and bottom-up algorithms to generate urban settlement patterns (Coates and Derix 2014). After the 1976 Space Syntax paper, Hillier concentrated on spatial analysis and no further concerted efforts were made between the two research strands of spatial analysis and generative design computing to inform each other.

The present research objective emerged between 2004 and 2007 from three events: A) the Politecnico di Milano design studio teaching in 2004 (Coates and Derix 2007) when the author developed a simulation for a morphological feedback system of urban growth and the generated states were perpetually evaluated by network analysis (see 8.1). At that time Paul Coates mentioned that Bill Hillier had aimed for such a solution where the generated dual network structure would evaluate the morphological syntax. B) the MSc Computing & Design teaching at UEL where the author provided students with genetic algorithms and an isovist analysis as the fitness criteria for GA generated plan configurations (see 7.1). And C) through conversations with Alasdair Turner with whom a paper was planned on a prototype about the discussion. Turner did publish a paper and acknowledged that *"this paper is based on an original idea by Christian Derix"* (Turner 2006, endnote).

The vehicle for the synthesis of spatial analysis and generative computation is provided by spatial configurations, which correlate generative algorithms to analytical representations. Very few individual attempts existed at this point but in order to make the correlation instrumental, a series of projects were required to generalize the interfaces between the two representations, particularly in relation to the user-centric component of analysis. Hence, the structure of the core case-study chapters distil in the first chapter a series of generative algorithms that focus on spatial organizations and the second chapter on a series of algorithms that analyse spatial properties in relation to human-centric behaviours and cognition. The third chapter illustrate the first isolated attempts at a correlation of the two and how they constrain each other with different levels of autonomy from the observer as designer. The levels of autonomy provide different potentials of embedding into a design workflow.

OBJECTIVE 2: DEVELOP A STRUCTURE FOR A CONCEPTUAL FRAMEWORK THAT BEST FACILITATES A COMPUTATIONAL SYSTEM FOR HUMAN-CENTRIC GENERATIVE DESIGN.

The intended computational system for human-centric generative space planning was aimed to be fit for *designing* that is, not for post-design analytical rationalization or for academic research purposes. Such as system must be useful and embeddable into a live design workflow while all its components are scientifically sound. The conventional segregation of the design process into *generate-and-test* or *analysis-synthesis* utilized for decades in computer science (Darke 1979) and a backbone of linear parametric modelling in architecture was not an option as it would not be able to allow insights into the correlation between heuristics and phenomena from a behavioural and cognitive legibility point of view. While practical applications commonly represent causally closed design automation for specific relations such as parametric associations in visual programming applications[3] like Bentley's Generative Components or McNeel's Grasshopper[4], and complex academic models aim to simulate all possible conditions thus locking down most interaction, a new model for spontaneously associating diverse algorithmic representations was required. Robin Liggett's technical survey including criticism of academic research and developments

[3] Visual text-fields called nodes represent procedures that are connected into a linear execution sequence.
[4] http://www.grasshopper3d.com, accessed 05.01.2015

(Liggett 2000) as well as Rittel and Webber's theoretical design methodology criticism (Rittel and Webber 1973) will be used to support the thesis.

Chapter two will discuss design process structures in relation to computation, touching on problems of openness of the design workflow and knowledge encapsulation. The case-study chapters four and five will discuss generative and analytical algorithmic design models. Chapter seven and eight describe strengths and weaknesses of their associative integration into spontaneous non-hierarchical models. Finally, also in chapter eight, the transition from an integrated algorithmic design model to an openly associated framework is discussed that culminates in the Open Framework for Spatial Simulation (OFSS).

The dissertation is not emphasizing technical descriptions of algorithmic behaviours, modalities of interaction or interface design at the technical level of computer science. Most of the technical descriptions of the algorithmic models used here are available for review in the public domain. While algorithmic concepts and some technical idiosyncrasies underpinning the embedding of computational techniques into generative and analytical design will be discussed, the emphasis lies on the development of a conceptual framework of algorithmic models and their synthesis in the context of human-centric space planning in architecture.

1.4 THEORETICAL AND ARCHITECTURAL BACKGROUND

The research objectives have been informed by architectural and theoretical approaches beyond the specific arguments of the dissertation and having influenced the work significantly without being elaborated. Some are briefly outlined here:

1.4.1 Architectural Paradigms

By standard assumption, all architectural design methodology should be user-centric when (human) clients are involved. But architects have often lost this focus to sculpting (fine art), engineering (tectonics) or management tasks (utility). Hence, a series of prominent human-centric space planning movements have reacted against the objectification of spatial environments. When those counter-currents occur, it becomes clear that the human-centric argument in architectural design appear to have little scientifically grounding.

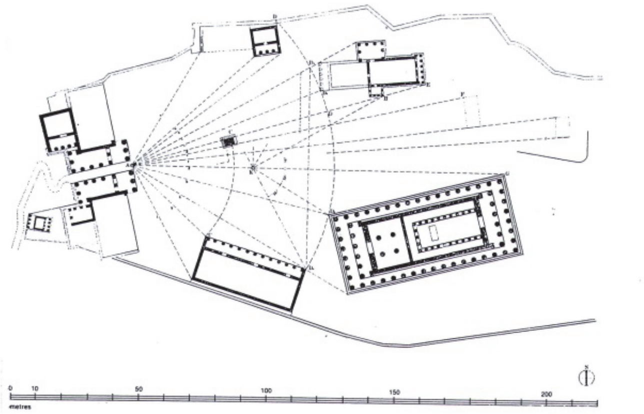


***Fig1**. Doxiadis (1972): view locations giving proportion to Acropolis*

Greek architects produced masterplans based on the perception of the structure of the urban environment. The ordering principle was based on the imagined civic user moving (*peripatetic*) through the spatial environment (Doxiadis 1972). Robin Evans explains how social status might have led to the evolution of designing complex movement structures, i.e. corridors and circulation: "*This split between an architecture to look through and an architecture to hide in cut an unbridgeable gap dividing commodity from delight, utility from beauty, and function from form.*" Evans speculated that *"as if from the architect's point of view all the occupants of a house […] had become nothing but a potential source of irritation to each other"* (Evans 1997, p8). Even though the correlation between space planning and occupancy control appears negative, it constitutes a human-centric advance in architectural design that became ever further entrenched until it peaked during the rationalist movement at the beginning of the 20th century when all human action was subjected to logistically efficient procedures.

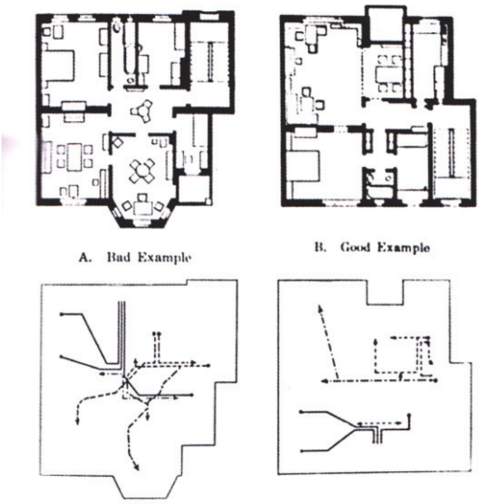


Fig2*. Alexander Klein, 1928: Functional House for Frictionless Living (Evans 1997)*

At the end of the 19th century, August Schmarsow paved the way for the German organic architecture movement that represented an alternative to Rationalist movement of the early 20th century. Schmarsow suggested that *space* is the generator of architectural form (*raumgestalterin*) and that space is isomorphic to human movement and perception, proposing a scientific correlation between spatial structure and user behaviour (Schmarsow 1894) . His concept of 'kinetic perception' represented a hybrid of contemporary German art and psychology theories which touched on spatial perception (Schwarzer 1991): Robert Vischer's *aesthetics* espousing a correlation between human physiology and objects; and Theodor Lipps' concept of *empathy* proposing a correlation between internal structures and their behavioural states of a cognitive subject and external objects in order to create shared knowledge (Derix 2014).

Hugo Haering elaborated Schmarsow's concept into the German organic architecture, declaring that space must be *essential form*, whereby essence was understood to be a system that accommodates the occupant: "*[…] a house designed on organic principles 'understands' its supporting role. It receives its gestalt from the inhabitant and his inhabiting [...]. It becomes wesenhaft (essential) and does not* *belong to the abstract form"* (Janofske 1984, p25), meaning appearance and geometry are subservient to occupation. Unlike Schmarsow, Haering was an architect and his evolution of Schmarsow's concept does not isolate the user-space correlation but links it to design methodology. An organic form (*organform*) had to perform on functional principles (*leistungprinzip*) that could be developed through a *gestalfindungsprinzip*, i.e. a form-finding principle. The *gestaltfindundsprinzip* based on functional user-centric performances encapsulated the correlation between user and form, which Haering called concordance (*konkordanz)*.

Hans Scharoun built on Hugo Haering's organic design theory and elaborated the *gestaltfindungsprinzip* by invoking a design approach called *improvisation* (Janofske 1984). Scharoun approximated user behaviours as if by mental simulation and he wanted to use this simulation to guide the design form. Hence, his definition of improvisation was that *„When the emphasis is occupation, behavior drives design and with it improvisation"* (Janofske 1984, p136)[5]. Scharoun's Berlin Philharmonie represents this inside-out design methodology well where no globally structured form can be detected but an assembly of locally performing spaces dictating the appearance.

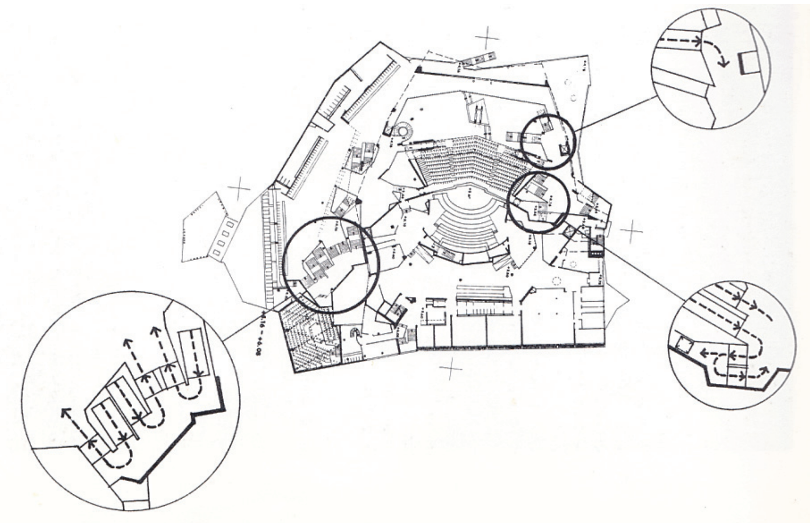


***Fig3**. Hans Scharoun, Berlin Philharmonie, 1963: the plan diagram illustrates how Scharoun anticipated relations between user behaviours and the configuration of geometry. He regarded space as an awareness-structure, not a material boundary (Janofske 1984)*

Although Herman Hertzberger is associated to Structuralism his modular structures were imbued with anticipated use patterns (Hertzberger 2005). Like Scharoun, Hertzberger designed through aggregation but unlike Scharoun, Hertzberger pre-designed a limited set of permissible spatial modules that in combination were presumed to give rise to social and use conditions. In his 1972 Centraal Beheer project, the site served as generator that informed the aggregation sequence. Each module was instilled with *valence*, which in combination with other modules produces affordances for occupation, creating *polyvalent* spaces from simple modules and a quasi-grammatical syntax (Hertzberger 2014).

[5] *„Wenn aber das Verhalten im Vordergrund steht, ist das Handelnde am Zuge und mit dem Handelden die Improvisation."* (original quote with author's translation in text)

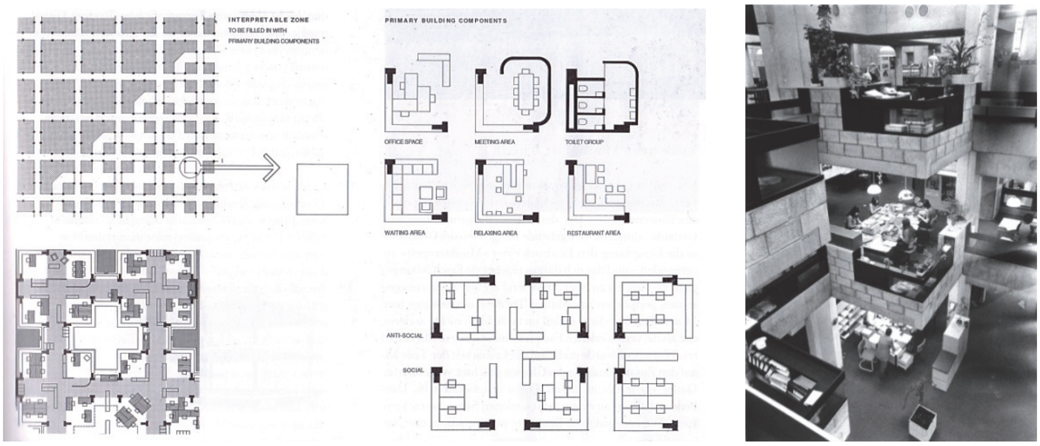


***Fig4**. Herman Hertzberger, Centraal Beheer office building, Apeldoorn, 1972: spatial units are combined into social and use conditions like a language grammar (left); and (right) an interior view*

Noam Chomsky's (1957) theory of syntactic construction of language and semantic meaning provided much of the basis for architectural theory such as Hertzberger's polyvalent structuralism and evolved further into Derrida's (1982) deconstructive theory of *languaging*. Apart from Hertzberger, Bernard Tschumi managed to translate these theories into human-centric design methodology, particularly with his design for the Parc de la Villette. Based on Derrida's deconstructivism, Parc de la Villette represents a self-organizing field of latent conditions that are activated by occupation (Coyne 2011). The project therefore constitutes one of the first spatial environments not based on design methodology but on real-time self-organization from dynamic networks of acting occupants.

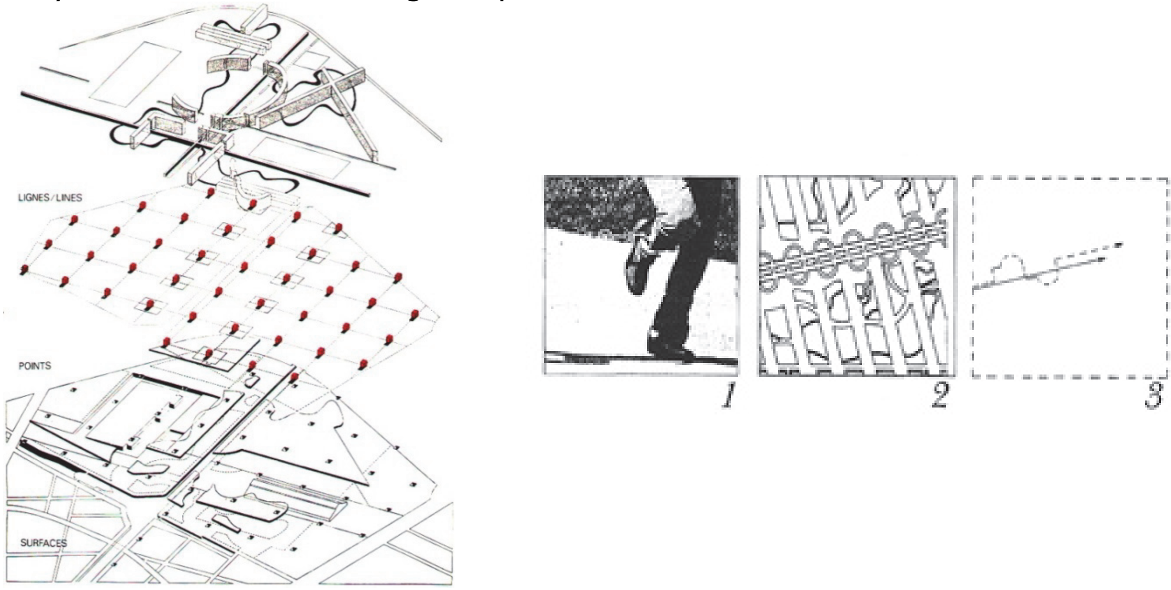


***Fig5**. Bernard Tschumi, Parc de la Villette, 1982: (left) the design methodology involved layering different notations to release control to an un-programmed system for occupation; in his Manhattan Transcripts publication (right), Tschumi (1994) always sought the correlation between action and form through generative diagrams*

When architects do not attempt to generalize their design methodology theoretically, human-centric design is often found to be reduced to the 'genius' of the architect. Architects such as Steven Holl design very tangible human-centric spaces but decline to theorize on their methodology and choose to eulogize the physical result instead, turning the discourse into intangible notions of phenomenology (Holl et al 1994).

1.4.2 Scientific Paradigms

The field of computational design is full of references to Norbert Wiener's theory of cybernetics, which was introduced to architecture in the UK by Gordon Pask (1969). Both John Frazer (1995) and Paul Coates (Coates and Derix, 2014) were at the Architectural Association when Pask ran a seminar on cybernetics and they became responsible for spreading cybernetic design thinking into the computational design community (such as Frazer's early collaboration with Cedric Price in 1978 for the Generator Project (Price 2003)). At the core of early cybernetics as the theory of control systems was the concept of feedback, which allows a mechanical yet systemic structure to adjust to variations of external parameters.

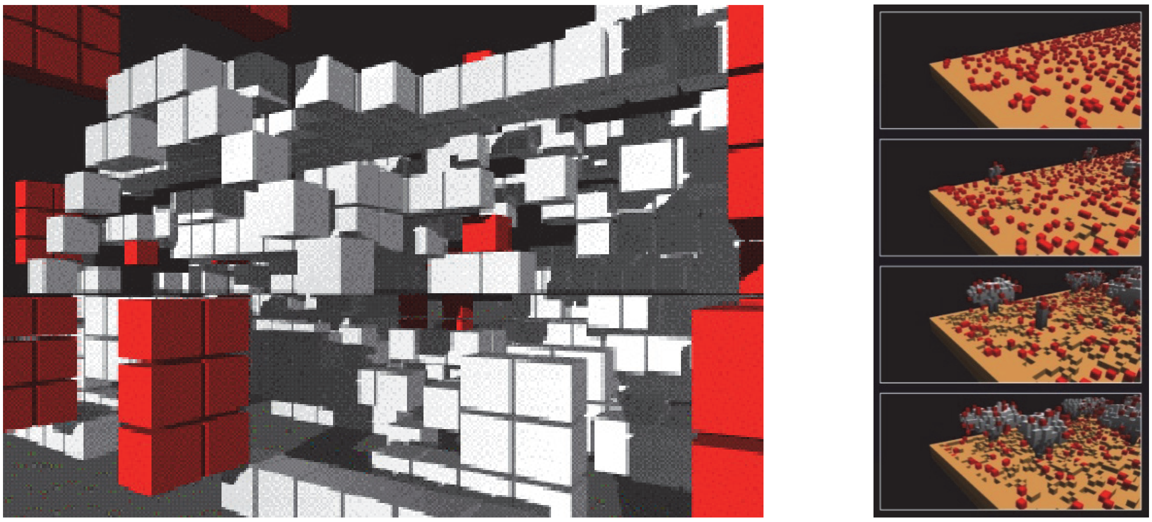


Fig6. *Paul Coates, 3D cellular automaton, 1996: Coates introduced the first volumetric cellular automaton to generate spatial configurations (left); Guy Theraulaz, stigmergic construction, 2014: Theraulaz encoded Pierre-Paul Grassé's stigmergic principles and simulated scenarios of animal construction (right)*

As such cybernetics was initially aligned with contemporary theories of structuralism such as Chomsky's syntactical linguistics where a system can produce infinite states but not invent new state spaces, which is referred to as first order cybernetics. This distinction was provided by Heinz von Foerster (1984) who introduced the notion of *cybernetics of cybernetics* to indicate that truly self-organizing systems need to be able to also organize their internal structure not simply react to external stimuli. Von Foerster's second-order cybernetics discussed meta-systems for cognition and asked the question of 'how to perceive' and understand the environment cognitively, referring to Humberto Maturana's (1970) biological theory of self-reproducing systems or autopoiesis. Maturana and von Foerster's theories of systemic self-organization share the concept of interaction between systems that are each structurally constrained but through their mutually affecting actions produce new state spaces, called *consensual domain*. Within such domains, two or more systems organize their internal structures so that it can sustain the interaction, called *structural coupling*. A consensual domain between systems represents a cognitive convention between observers who adjust their internal structure unconsciously. Hence, while first order cybernetics was a theory of mechanical control, second order cybernetics is a theory of cognitive organization.

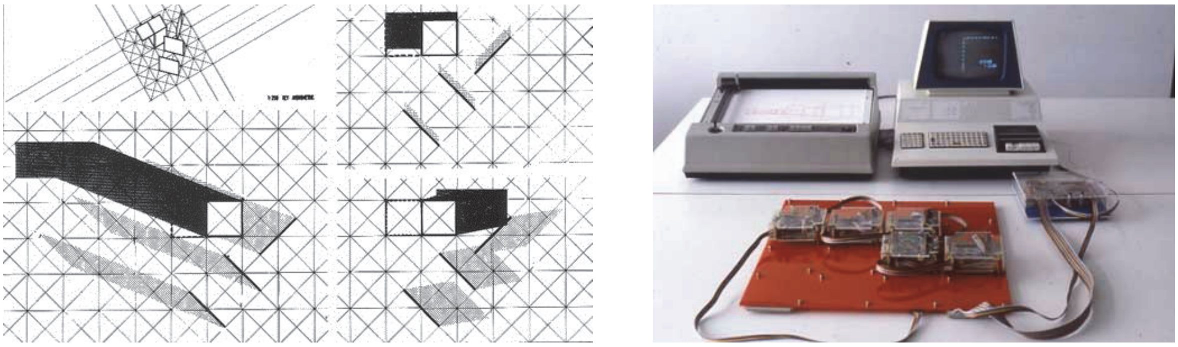


***Fig7**. Cedric Price, Generator, 1978: (left) a plan view of physical partitions on an environmental grid at a certain time; (right) Julia and John Frazer's computational prototype of the Generator system which aimed to simulate autonomously guided hardware modules representing the spatial partitions (Frazer 1995)*

This transition changed the focus of attention from machines to humans and their environments, to human and ecological epistemology. The development owes much to developments in computing science starting with Stanislav Ulam and John von Neumann's automaton theory (Langton 1995). The model for automaton theory is the cellular automaton (see 3.4) which had its breakthrough with John Conway's Game of Life, demonstrating intelligibly how pattern and phenomena emerge from syntactical field conditions. As a consequence, automata inspired both artificial intelligence (AI) and artificial life (AL): AI due to the computational representation of knowledge and operation; and AL due to environmental and time-based qualities of social behaviour (Derix 2008). While AI informed the work of this dissertation through models of artificial epistemology such as artificial neural networks (ANN), which help to understand cognitive associations within data sets, AL provided most of the bases for perceptual and behavioural models discussed in this dissertation. But essentially, both AI and AL developed parallel processing systems that attempt to relate different pattern into complex feedback systems which decode some form of correlation between human and environment. Paul Cilliers' (1998) description of generating knowledge in complex (neural) AI systems is very similar to Abraham Luchins' (1968) description of cognitive processes of Gestalt theory:

"Some patterns will catch others in their wake in the sense that they will start appearing in concert. This process increases the order in a system and facilitates the formation of associations through resonance." (Cilliers 1998, p95)

"Wertheimer analysed this process as 'the motion is due to a field of activity among cells,…not excitation in isolated cells but field effects.'" (Luchins 1968, p525)

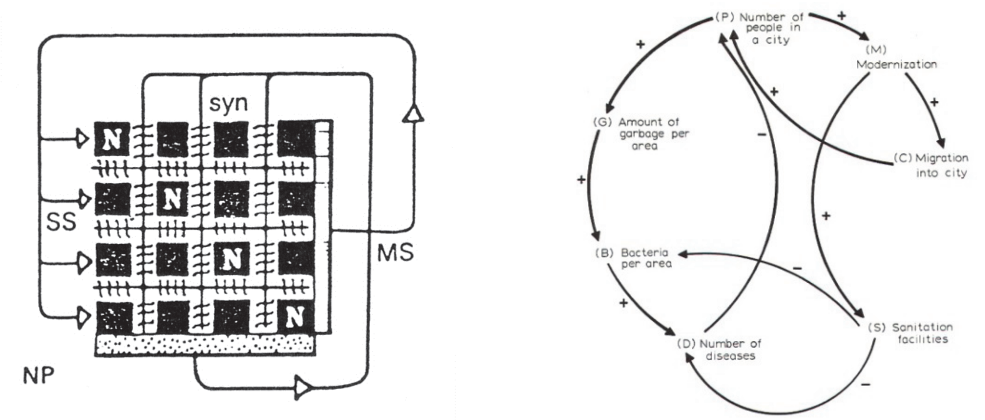


Fig8. *Heinz von Foerster, second-order cybernetics (1984): von Foerster's feedback system alters the structure and the weighting of a system and allowing for new cognitive states; Magoroh Mayurama (1963), deviation amplification: Mayurama introduced the notion of positive and negative feedback in systems and applied the concept to urban systems*

On the other hand, AL did introduce distinctions between actors not present in early AI. Cellular automata and artificial evolution provide ontologically important foundations for architectural simulation, such as representations of observers, agents and field (cells). This distinction introduces the opportunity for more complex models for architectural design simulation, differentiating between designer, occupant and space. Cognitive theories that initially relied on analogue models such as James Gibson's *ecologic perception* (1950) or Jean Piaget's developmental psychology could now be systematized into syntactical models (Piaget and Inhelder 1956). The core concept of *enaction* proposed as foundation of cognition by Gibson, Piaget and Maturana proposed direct interaction with spatial environments to be the source of spatial knowing[6]. Environments therefore provide intelligence to the observer in the form of *affordances* through *natural vision* (Gibson 1979). Affordances are perceived by observers through perceptual properties of their surrounding and enable decision about behaviours. This empirical approach opposed the then predominant theory of cognition being a segregated Cartesian cerebral activity and provided fundamental concepts to AL.

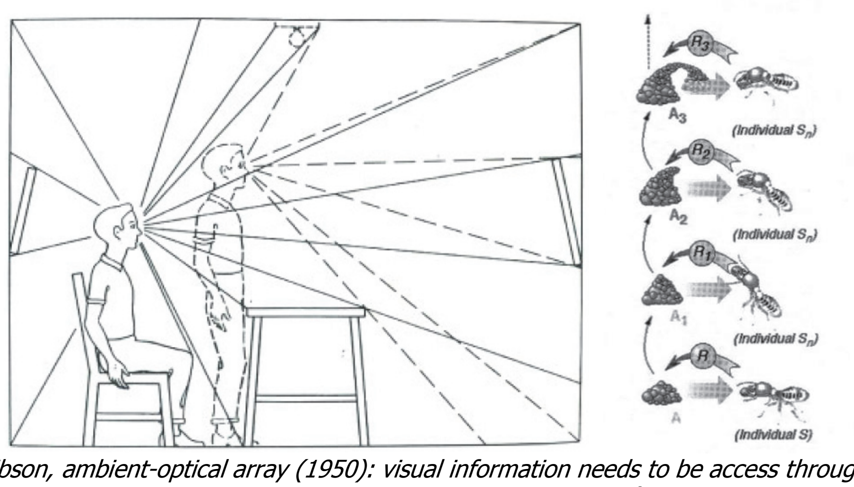


Fig9. *James Gibson, ambient-optical array (1950): visual information needs to be access through movement as the environment already stores cognitive structure (left); Pierre-Paul Grassé, stigmergy, 1950s: social animals communicate indirectly via traces in the environment that they leave to instruct others (Theraulaz 2014)*

[6] *"The intuition of space is not a 'reading' or apprehension of the properties of objects, but from the very beginning, an action performed on them."* (Piaget and Inhelder 1956)

As a former student of Piaget, Seymour Papert (1980) introduced the distinction into observer, agent and field (array of patches) in his Logo software to help children learn the logic of algorithms. The importance of the distinction lay not only in the behavioural separation of observer and occupant but in the allocation of intelligence in the environment. Gibson and Piaget understood interaction to be important but Papert's CA-based Logo software introduced the concept of an autonomously processing context. Papert's turtles behaved liked software versions of Grey Walter's (1950) hardware agents Elmar and Elsie and became the foundation for agent-based modelling (Langton 1995). Craig Reynolds (1999) and Valentino Braitenberg (1984) continued the development of situated or *embodied* agent-based environmentally behaviour. Reynold's *steering behaviours* of *boids* (bird-oid) provides a conceptual archive for many particle-based pedestrian movement simulations. The biologist Guy Theraulaz translated Pierre-Paul Grassé's 1959 theory of *stigmergy* into computational simulation which shows how social behaviour of animals is coordinated indirectly via the environment as a communication facilitator and instructor (Theraulaz and Bonabeau 1999). Similar to city growth, complex natural constructions by animals are often driven by environmental information deposited over time, rather than subjective intentions (Theraulaz 2014).

Despite the superficial difference, the core subject of AI and AL concerns 'communication' between systems or actors as a means to generate knowledge and phenomena, represented via systemic qualities such as distribution, parallel processing and feedback (Cilliers 1998). Spatial cognition and architectural design paradigms like Schmarsow's *kinetic perception*, Scharoun's *konkordanz,* Hertzberger's *polyvalence* or Tschumi's *superimposition* are supported by well formulated scientific models that explain the human-space correlation of their conceptual approximations.

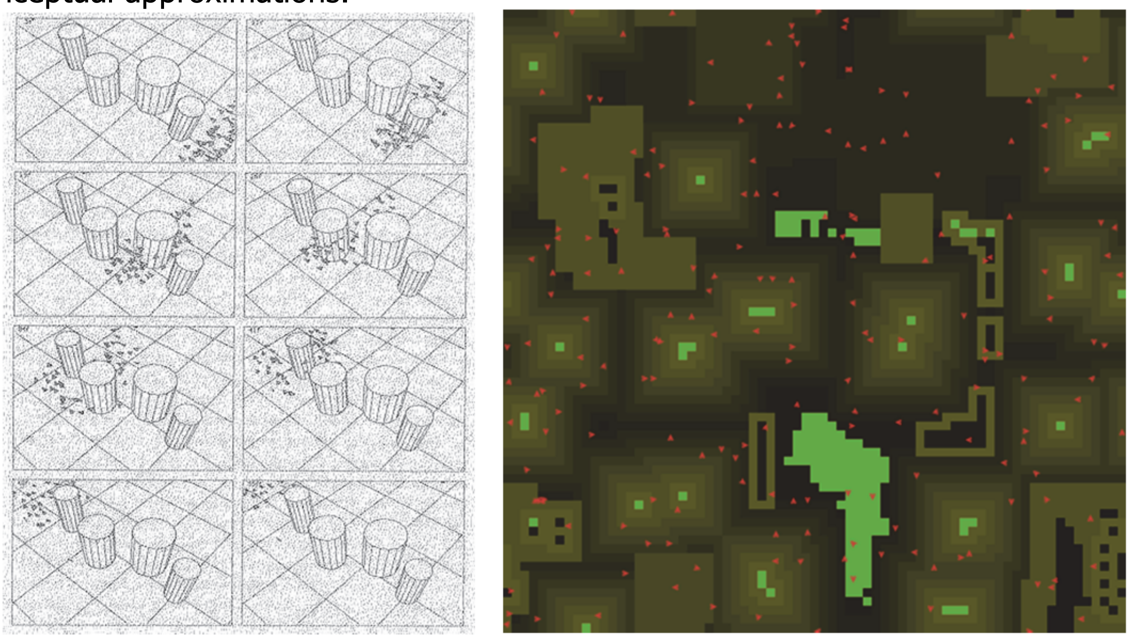


Fig10. *Craig Reynolds, 'boids' flocking behaviour (1999): stills of the flocking boids representing the behaviour of a swarm (left); Netlogo based on Seymour Papert's LOGO (1968): a field of patches that calculate neighbouring patches for values like a cellular automaton is used by 'turtles' – the red arrows – to navigate and deposit information back into the environment (Papert 1980)*

1.4.3 Architectural Theory

Whereas 1.4.1 highlights architectural design thinking that informed the design of human-centric spatial systems, this section briefly outlines architectural research that supported the dissertation.

As per the previous conclusion, scientific architectural research exists that underpins some of the architectural paradigms. One of the two core theories supporting this dissertation is Bill Hillier's original generative theory of spatial systems, called Space Syntax (Hillier et al. 1976). Particularly, the notion of the *field* plays a major role in many 20th century theories of architectural space. Hillier interpreted the field as an abstract layer of performances that represents active and passive structures such as the actual spatial configuration and its meta-structure revealed through occupational patterns. He called the meta-structure the 'generic function' of a configuration where user occupation and spatial cognition correlate to geometric conditions. In the chapter Laws of the Field in Space is the Machine (Hillier 1996) unknowingly supports the above mentioned architects by stating: *"Generic function refers [not to ...] but to aspects of human occupancy of buildings that are prior to any of these: to occupy space means to be aware of the relationships of space to others, that to occupy a building means to move about in it, and to move about in a building depends on being able to retain an intelligible picture of it".* (Hillier 1996, p284)

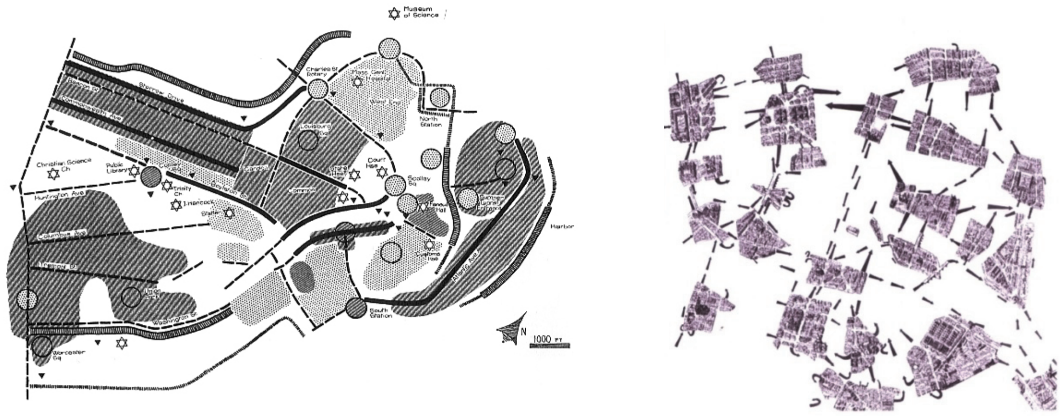


Fig11*. Kevin Lynch, mental map of Boston (1960) (left) and Guy Debord, psycho-geography mapping of Paris, 1955 (Sadler 1998)*

Beyond the lesser known work of Hillier, there are other more popular architectural theories of spatial fields and user-centric correlations. From a planning perspective, Kevin Lynch's (1960) research into the cognitive structure of the city from an occupant's perspective is well known. Lynch's *mental maps* approximated the cognitive organization of urban environments deduced from people using their neighbourhoods. Lynch distils the configuration of a city into five geometrical elements (paths, edges, districts, nodes, landmarks) and defines their perceived interrelations, also employing Gestalt theoretic principles to explain holistic associations. The key insight in the formation of this dissertation's research question was his demonstration that geometric representation alone does not provide a cognitive standard for legibility of environments and that non-architectural methodologies are required in the explanation of the correlation between users and spaces.

Similarly, the psycho-geographic mappings of the Situationists of the 1950s around Guy Debord inspired the application of network-based representations from a user perspective (Sadler 1998). As the name *psycho-geographies* implies, subjective representations of urban fields were developed based on objective methodologies. The underlying principle of most mappings was called the *dérive* (or *drift*) which describes rules for local tactical decisions for way-finding rather than global strategies. Those local tactics dependent on contextual events produced global subjective maps of specific users or uses that break with conventional Euclidean city plans.

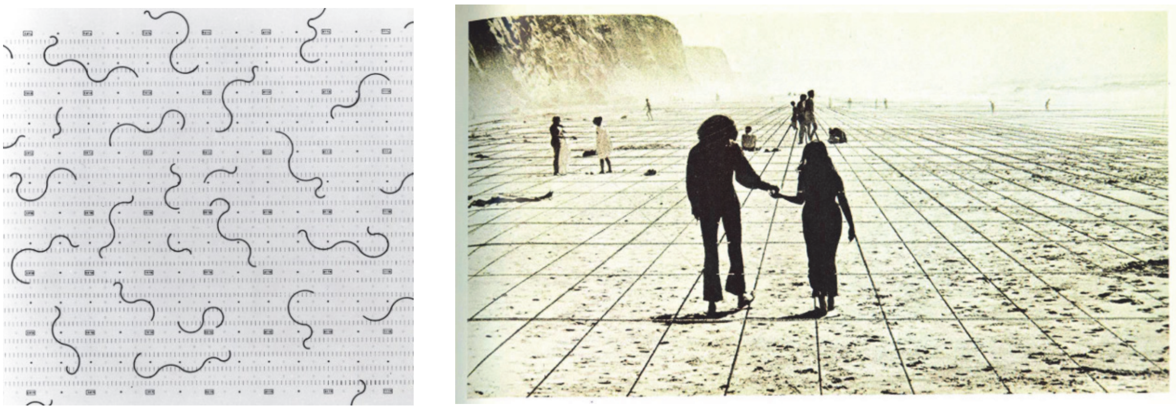


Fig12. *Archizoom, No-Stop City, 1969 (left) and Superstudio, Supersurface, 1972 (right)*

More conceptual analogies of the information field as space proposed by architectural design theorists of the 1960s inspired the aesthetics and also the human-scale representation of the field. Born out of an anti-authoritarian *zeitgeist*, new models of self-planning spatial environments were thought up such as Archizoom's Continuous Habitation (Branzi 2006) or Superstudio's Supersurface (Lang and Menking 2003). Their concepts used discretization of space analogue to machinic processes or computational representations of grids. In kinetic art on the other hand, Gianni Colombo developed interactive installations such as Spazio Elastico (Scotini 2006) conveying computational theories such as cellular automata or Konrad Zuse's Calculating Space (1952) into architectural diagrams of space as an interactive autonomous system. Particularly, Spazio Elastico demonstrated the notion of mutually perturbing systems of users and environment, reflecting not only computational representation but also biological theories of cognition such as Maturana's *autopoesis* or Grassé's *stigmergy*.

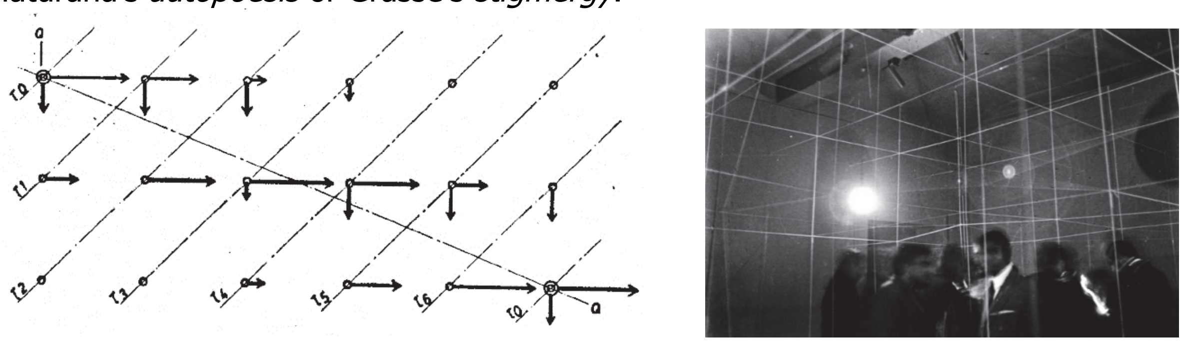


Fig13. *Konrad Zuse, Rechnender Raum, ('calculating space'), 1952 (left) and Gianni Colombo, spazio elastico, 1967 (right)*

These early conceptual diagrams of interactive spatial systems were encoded by more technical research projects within the same *zeitgeist* of authorless self-designing fields such as Nicolas Negroponte's Grope (Negroponte 1970) or the above mentioned Generator project by Cedric Price with John Frazer. Grope represents just one of Negroponte's many interactive machine projects but well illustrates the autonomy of simply rule-based users (albeit as software or hardware agents) to operate in information rich fields generating new environmental states through their interaction. Regarding many of those conceptual diagrams, due credit for most network and field like representations must be given to the discipline of geography, whose representations and local mapping techniques precede architectural research. This is well summarized in Haggett and Chorley's Network Analysis in Geography (Haggett and Chorley 1969).

The authorless dynamics of the field giving rise to conditions that serve architecture to 'become what it wants to be' is what Colin Rowe called the *maximum non-interventionism* (Rowe 1983). This approach evolved and became generalized until the end of the 90s when procedural mapping and diagrams had their peak illustrated by James Corner's summary of mapping techniques in The Agency of Mapping (Corner 1999). Stan Allen's essays on *field conditions* and the evolution from *object to field* (Allen 1997; 2008) reflect on those trends and provide a bridge from the machinic to the digital and syntactical analogy and link the field representation to architectural design without the previous political critique (discussed in following chapters).

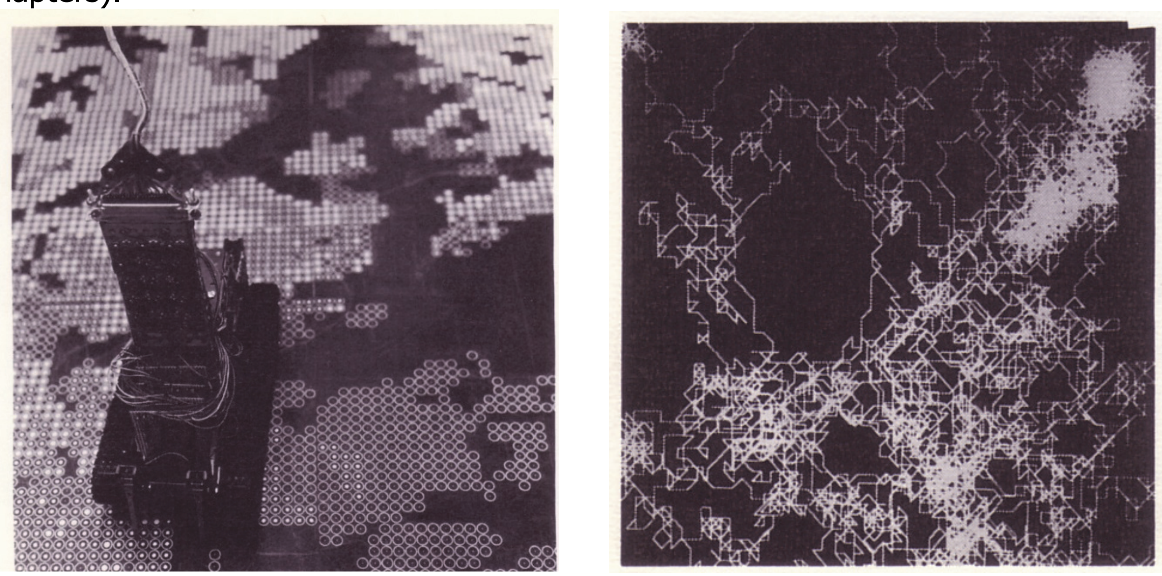


***Fig14**. Nicholas Negroponte, GROPE, 1970: sensor equipped robot reading map (left) and output map (right)*

1.4.4 Architectural Practice – emerging human-centric value of space planning

As an academic researcher embedded in architectural practice and confronted with live projects on a daily basis, the author has noticed how practice has partially moved further into the direction of user-centric space planning than academia. While academia has attempted to explore computation as a formal or efficiency tool, many professional design guides of various sectors have adapted their approach from quantitative target compliances to key performance indicators, many of which revolve around the user as a benchmark. An early example is reflected in the

Transport for London station planning guidance which was adjusted after the King's Cross fire where 31 people died due to a lack of cognitive support for way-finding of the station's configuration, which was purely optimized for direct connections not for orientation (Phillips 2004). All new stations of the Jubilee Line extension of the late 90s needed to provide natural way-finding conditions to facilitate user perception.

***Fig15**. Roland Paoletti, Jubilee Line station Canada Water, 1999*[7]

Urban planning has been revolutionized in the 90s, led in the UK by the Council for Architecture and the Built Environment, culminating in the UK leading urban design guidance called ByDesign (CABE 2000). ByDesign compiles urban design objectives and aspects of development form as discursive compliances and KPIs which are reminiscent of Kevin Lynch's perceptive elements and Hillier's space syntax analysis. The seven objective headings include *character* as pattern, *continuity and enclosure* as geometric properties, *quality of public realm* as subjective sensation, *ease of movement* as accessibility performances and *legibility* as structural organization (CABE 2000, p15), i.e. mostly based on qualities dependent on human cognition and behaviour.

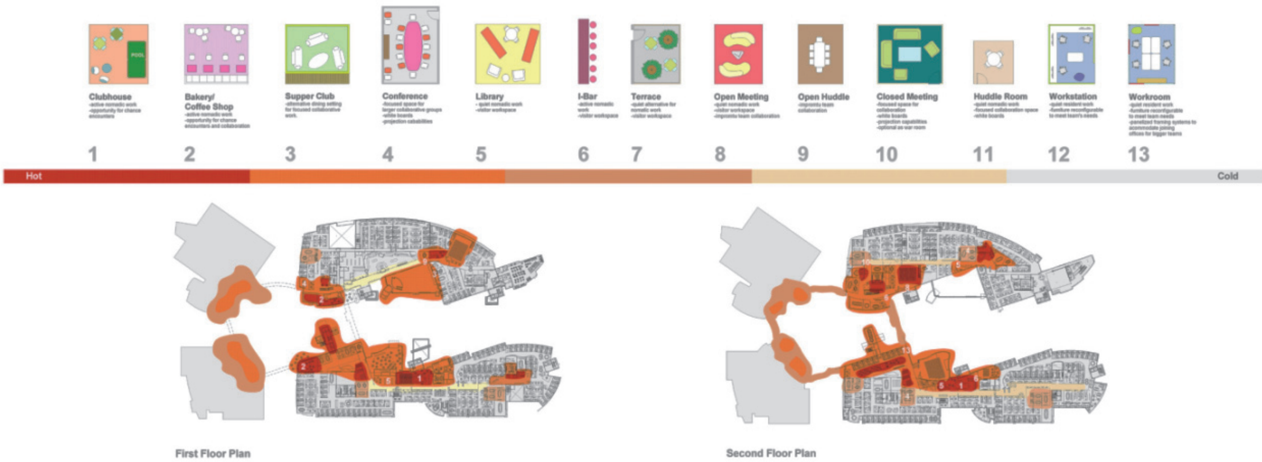


***Fig16**. Clive Wilkinson architects, GooglePlex, 2004: user-centric and activity-based workplace design for Google*[8]

[7] © Ben Brooksbank, accessed 23 February 2014

Also private building sectors are affected such as commercial workplace design described in WorkPlace Matters by the US General Services Administration (US GSA 2006) or the British Council for Offices' Best Practice in the Specification for Offices (BCO 2009), to name but two. In order to increase productivity, workplace environments are developed currently by occupant activity patterns rather than area specifications. Further social interaction is fostered directly by design to promote communication between staff to increase information exchange as a means for efficient project management. The 2009 BCO Best Practice in the Specification for Offices states: *"Conventional North American, British and Japanese office developments have sought to minimize circulation space to achieve maximum efficiency. Many northern European examples on the other hand demonstrate the importance attached to the function of circulation areas as interactive and stimulating communal workspaces which promote a sense of amenity and enhance working effectiveness and productivity" (BCO 2009, p52)*. The underlying drivers of this new design approach are user behaviours and perceptions that bind isolated functional programmes into proto-urban experiences. Again from the BCO: *"Staircases to be located for building users to have the option to use them over lifts, with the design of the staircases providing visual connection and social interaction opportunities."* (BCO 2009, p44)

The same user-centric design focus is evident in hospital planning, education buildings, airports, convention centres and others. The new approach displays an awareness that better spaces must be based on user experiences and their relation to spatial organisation, on social interaction and human cognition as drivers to achieve better performances. This is not however a romantic or selfless change as the profession and its clients have realized that added value depends increasingly on the distinction of place by pleasant experience and adaptability, which is ByDesign's sixth objective.

1.5 METHODOLOGY & STRUCTURE

The argument of the dissertation is constructed from historical research in the field of design computation, using literary sources as references. The development of corresponding aspects of the argument is then illustrated using case study models. Eventually, diagrams help to visualize the structure of the OFSS, which represents the current state of research used as a temporary outcome of the research questions (temporary because the framework is in continuous development).

1.5.1 Case Studies

All case study models described in chapters 5-8 were either designed, developed, supervised or procured by the author. Case studies from the University of East London's (UEL) Centre for Evolutionary Computing in Architecture include the author's own master thesis and models developed by students supervised by the author. Additionally, most models of CECA are based on code provided by the author

[8] http://www.clivewilkinson.com/pdfs/CWACaseStudy_GoogleplexANewCampusCommunity.pdf, accessed 23 April 2015

and scopes issued to students with example code snippets. Case studies from the Computational Design Research group at Aedas architects (CDR) have been developed with several members of the group[9]. The group has been founded, developed and managed by the author. It has now moved to WoodsBagot architects and is called SUPERSPACE.

Case study models are drawn from various settings including academic developments from student exercises and master theses or professional developments such as competitions, scheme design stages, strategic planning, commissioned design software, in-house research and collaborative research.

1.5.2 Omissions

Apart from chapter two *Models of Artificial Design* there is no explicit chapter on precedent work discussing similar computational models. Chapter two discusses computational approaches rather than discrete models to define the dissertation argument. Specific computational models of space planning are referenced within the core case study chapters five to eight. No comprehensive body of work referring to computational human-centric space planning exists for reference. Particularly, no distinct developments like the resultant OFSS exists that synthesize the three strands of generative design, spatial analysis and design heuristics that could be referenced.

Purely generative computational models for form-finding like George Stiny's shape grammars (Stiny 1972) or form such as Greg Lynn's work (Lynn 1999) are not the subject of the dissertation and thus excluded. Many experimental generative computational models have been developed and published since 2005 such as the models of Axel Kilian's PhD (Kilian 2006) that are also excluded for their emphasis on engineering or formal aspects of architecture, ignoring spatial configurational and occupational properties.

Similarly, there is no exhaustive chapter on the technical specifications of algorithmic models. Chapter three gives a brief overview of some fundamental algorithms used in the case study chapters five to eight. The focus of the dissertation rests on the conceptual argument of how to develop a USOM rather than a precise syntactical description of the code. Most algorithms and mathematical foundations employed in the case studies are in the public domain and each case study discusses the modification of the basic algorithm in order to arrive at the specific representation.

Finally, while many easily usable application programming interfaces (API) with graphic user interfaces (GUI) have been made available since 2005, all interaction and interfaces have been developed by the author and colleagues independently. One key omission in relation to technology lies in the description of what is called interaction design. Interaction design provides an important ingredient to the computational framework here presented but its code specifications are neglected to give more space to the algorithmic representation and conceptual narrative.

[9] Members of group over the last 11 years whose models are featured are: Pablo Miranda, Åsmund Izaki (previously Gamlesaeter), Lucy Helme, Prarthana Jagannath and Anders Holden Deleuran

2 MODELS OF ARTIFICIAL DESIGN

The understanding of a computed or artificial model of design has not been stable since the advent of computers in design disciplines. Both epistemologically and ontologically, the objectives and specifications have varied based on the separation of tasks that specifies the relationship between the 'machine' and the human as user. An envisaged USOM must interface knowledge and process components that have changed their roles in the organization of computational models over time. The principal components are the constraint set (brief and design requirements), the field of conditions (context) and organizing process (design heuristic and algorithmic model). Hence, the discussion in this chapter revolves around the user-machine relationship from an organizational perspective as well as epistemological and ontological paradigms of artificial models of design, particularly architectural and urban design that provided foundations in the development of this dissertation. It omits architectural research such as William Mitchell's *Logic of Architecture* (1990) that has not had a direct impact. Similarly, developments from related fields where relevant developments have taken place like geography, arts or engineering have been omitted unless explicitly listed.

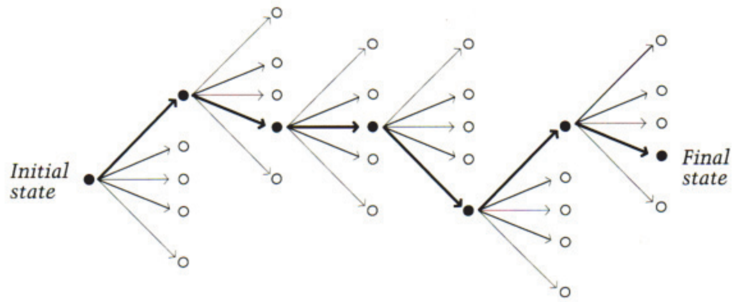


Fig17: *William Mitchell (1990): diagram of design process as linear decision sequence (generator path)*

2.1 ARTIFICIAL

The research of design as epistemology (process knowledge) and ontology (process objects) has its roots long before the 20th century as craftsmen and early architects of the renaissance built machines and produced representations via abstraction methods like projective geometry. Both require a level of abstracting knowledge and objects through which to translate abstractions and perform transformations on or through (Evans 1997). But it could be argued that the advent of the computer as a universal simulation machine allowed the quest for the representation and processing of external objects to extend to abstracting and simulating internal states, meaning human cognitive processes. This kind of abstraction requires representations that can be processed and hence formal descriptions. In that sense the artificial is not only restricted to externalized man-made artefacts but also to internal human states.

This chapter reviews the last 50 years of formalization of the design process from both the processing model as well as the types of knowledge that are processed. As computable formalizations were increasingly required, new models for external, internal and intermediate (interfacing) state representations emerged. Clearly, representational models are bound by the hardware of current computers, meaning the representation and transformations of symbols by the currently most accepted computer architecture and change according to its media.

2.2 DESIGN AS THE SCIENCE OF SEARCH

Although Christopher Alexander's Notes on the Synthesis of Form (1964) could be regarded as one of the first systematic descriptions of the design process rooted in Operations Research (OR), it was Herbert Simon's book The Sciences of the Artificial (1969) that built the foundation for the Design Sciences and provided the 'design by computing' research community with most of the essential concepts and terminology.[10] This might be because Simon was not a designer but a scientist who managed to transfer his many research insights from artificial intelligence, management and cognitive science into a theory of the design processes of man-made objects. As a behavioural and organizational scientist Simon together with Alan Newell had studied human behaviour and wrote extensively about heuristics – 'rules of thumb' or decision sequences – that seek to attain goals (Simon and Newell 1958).

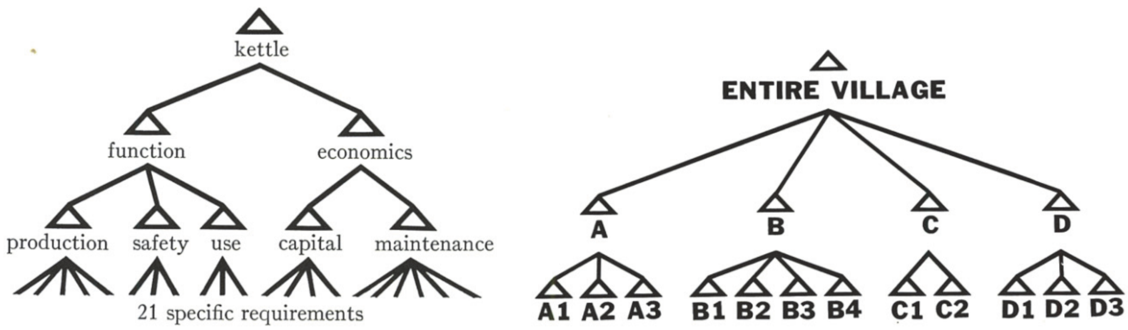


Fig18. *Christopher Alexander (1964): like an engineering problem, urban planning should approach townplanning by analytical decomposition into discrete functional units that are linearly linked like a tree*

In Sciences of the Artificial, Simon describes design as a *search for alternatives* (1969, p27). This already hints at the change of perception from earlier OR: that a search is not a deterministic activity but an exploration of options for which the exact outcome is not necessarily known *a priori.* He anticipates later criticism of scientific formalisms such as Philip Steadman's critique (1979, p196) of Alexander's static description of the analysis and synthesis of sub-systems through a single phase by showing that the so called *generator-test cycle "need not be a single cycle*

[10] The Stuttgarter School around Max Bense and Abraham Moles in the France had already experimented with computational design methods primarily for the arts and aesthetics at the beginning of the 1950s. Bense started his 'Information Aesthetics' in the context of the field of semiotics and Moles in response to Norbert Wiener's cybernetics and Claude Shannon's Information Theory (Kluetsch 2007). Students of the Stuttgarter School like Frieder Nake or George Nees developed early stochastic generative designs at the beginning of the 1960s on Siemens computers and even went as far as generating swarm designs that they not only plotted on Zuse plotters but also constructed 3D prints from (ZKM 2004). But Bense's philosophy of information aesthetics, even applied to architecture by Manfred Kiemle in the mid-60s, became esoteric and stalled. The work by his school into stochastic design processes akin to the Art Concrete movement could however be regarded as the first ever generative computational designs.

but there can be a whole nested series of such cycles" (Simon 1968, p 129). Therefore, a design cannot merely be understood as a problem-solving activity whereby a problem is assembled from goals but a search for appropriate assemblies (Simon 1968, p 124). According to Simon, in the search for alternatives success depends on the learning of associations between *states* and *actions*. States are configurations of the generators and to reach a problem definition, the designer has to learn the *generation path* that can produce such a *goal state* from an *initial state*.

The generation path represents a series of transformations (actions) that reduce the differences between initial and goal state, keeping in mind that a goal state is not necessarily known beforehand but rather a satisfactory configuration of the artificial system (artefact) that results in the mediation between *inner* and *outer* environment (Simon 1968, p 6). Clearly, this requires a notion of what the artefact is meant to mediate, i.e. its purpose, but does not propose how to attain that mediation state. This goal-attaining state could be one amongst many permissible configurations, which make up the *state space*. All goal-attaining states represent behaviours responsible for an artefact's agency, which means that certain configurations create specific domains of agency. The configuration represents an association structure (*table of connections*) that defines the generator (Simon 1968, p 122) and as a mediator between inner and outer environment, the artefact represents an interface.

As it cannot be known if a found 'good' state, which performs an adequate agency is the only possible optimal state, Simon talks of *satisficing* as a goal attainment strategy rather than *optimizing* as OR did. This contrasts with Alexander's (1964) notion that a set of design problems requires an *a priori* set of definitions for which the behaviour can be designed if the definition has been stated. Simon's *search* is a forward search that configures the state, while Alexander's is a *backward* search that configures the generator to linearly attain the set goal.

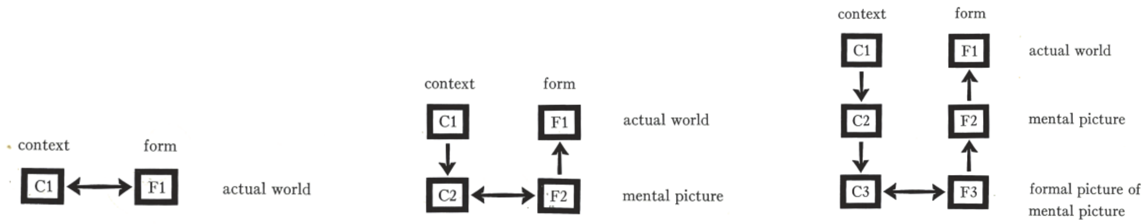


***Fig19**. Christopher Alexander (1964): Alexander claimed that architects create false pictures of their context and new abstractions are required via mathematics to create systems of design*

2.3 PROBLEMS WITH EXPLICIT REPRESENTATION

The definition of the design process by Simon and Alexander were operationally and systematically well defined. In Alexander's model, the state space typically represented exactly one state that could be backwards decomposed into many sub-problems to be solved while in Simon's model many alternative states were possible by forward search. Simon's model worked on the hypothesis of associative tables between actions and states that represented the reduction of differences between initial, current and target states. The aggregate set of reductions recorded as associations between past and future states made up the generator paths.

Simon further proposed the artefact to function as interface between internal environment – specifications of the brief – and the external environment – the constraints. Both Simon and Alexander assumed that the external environment and the resulting constraints for the internal environment contain all necessary knowledge to define the state spaces, and are therefore explicitly knowable and describable.

Simon's example of the General Problem Solver(GPS) (Simon 1968, p 123) set in a maze to find good paths between a current location (state) and a better location (state) illustrates the concept of the observer evaluating explicit action sequences (motor sequence) – the generator – that connect two locations in a design process. The example also regards the maze to be the external environment from which the constraints for the generation of the path can be deduced, proposing that the environment guides the designer.

Alexander's definition of *good fit* (1964, p 28-45) as correspondence between a well-designed artefact and its environment intended causally choreographed feedback between object and context where the object is a reflection of contextual dynamics and parameters. Simon relaxed this tight fit proposing that the specification of an artefact does not have to mirror the outer environment. In fact, the inner environment of an artefact needs to be *insulated* from the outer environment (Simon 1969, p 8-9). The insulation represents the design of the artefact itself and relays just enough but not exhaustive information for the inner environment of the artefact to work under changing conditions. It is slightly misleading then that Simon named the outer environment *mould*, meaning an exact positive set of information to reproduce negatives from, turning the artefact into a passive imprint of a specific condition.

Philip Steadman (1979) showed how the biological analogy of natural environment as design driver can be misunderstood. While a biological analogy based on Darwinism should have helped to foster the notion of non-teleological search, it has led to a rational design approach by which an exhaustive description of the outer environment would lead to a deterministic definition of the inner environment (*good fit*), transferring design purpose from designer to environment (Steadman 1979, p 198). Steadman pointed out how an artefact is in fact meant to *resist* environmental forces and mediate many different domains rather than accommodate exact environmental conditions.

A design problem and its process that might lead to a satisfying solution can be very complex and has to mediate many domains, making it difficult to formulate a single goal. Horst Rittel and Melvin Webber (1973) argued in their paper Dilemmas in a General Theory of Planning that urban planning as a design profession faces *wicked* rather than *benign* problems. They pointed out that professions such as architecture and planning had become perceived as sub-sets of engineering. Engineering problems are benign with a clear hierarchical solution path: "*define problem, gather information, analyse information, synthesize information and wait for creative leap, work out solution*" (Rittel and Webber 1973, p162). For urban planning as a social science it is difficult to implement systems theoretic and cybernetics approaches because the limits of the design context – the outer and inner environment – are hard if not impossible to define. They advocate a new or "*second generation systems approach based on a model of planning as an argumentative process in the course of which the image of the problem and the solution emerge gradually among participants as a product of incessant judgement*" (Rittel and Webber 1973, p162).

A decade later Donald Schön (1983) came to a similar conclusion as he investigated the design process of professionals. He criticized the models of Alexander and Simon as being too scientific in their approach to urban and architectural design, which do not possess *well-formed* problems like the sciences (Schön 1983, p47). He observed how professionals form design procedures iteratively by reflecting on actions, calling the design search *reflection-in-action* (Schön 1983, p73ff). Furthermore, he described how architects converge towards analogies that serve as *generative metaphors* through *reflective conversations with a situation* (Schön 1983, p185). This iterative convergence through reflection and conversation towards a problem definition is what Rittel and Webber (1973) would have called *re-solving conflicting ends.* Schön believed that technology was perceived for specific *ends* (1983, p41).

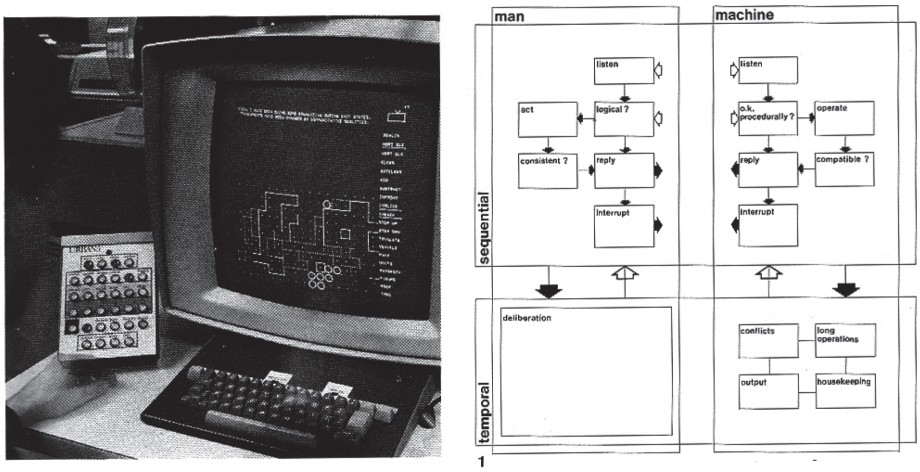


***Fig20**. Nicholas Negroponte (1970): temporary and sequential processes interfacing user and computer like a conversation on the Urban5 project*

The dilemma with the Design Sciences of the first generation systems approach was not just manifest in the misunderstood nature of the design brief but also in the fact that design disciplines lack clear target states. The previous encapsulation of design drivers through the specification of perceived objective inner and outer environments, i.e. constraints and ends, was now shifting to the design knowledge of the observer who directs the argumentative process or generative path. The argumentative process is an iterative re-solving of the problem-target definition. One of its aims is to converge towards a concept that can guide the process. Rittel and Webber (1973) speculated that a concept emerges through dialogue from learned knowledge and associations: "*one cannot meaningfully search for information without the orientation of a solution concept; one cannot first understand, then solve*" (1973, p162).

Both Simon and Rittel believed that *wicked problems* are not describable through quantitative absolute targets but through observation. The observer requires knowledge about the model that a simulation is representing; he needs to understand when a state is a good state beyond simple quantitative target satisfaction. To resolve this soft targeting definition, Simon borrowed the term *aspiration level* from psychology (1969, p30). Because dimensions in multi-variant search cannot be compared, qualitative benchmarks such as aspiration or least dis-satisfaction are introduced. When a solution is 'good enough' it *satisfices* aspiration (Simon 1969, p 30). It cannot be true or false just good or bad (Rittel and Webber 1973, p163). In computing, most simulations are programmed with a halting rule that indicates when a state has been reached that complies with all performance criteria. For wicked problems, it was argued, a halting rule cannot be defined.

2.4 AUTOMATING KNOWLEDGE

Search and dialogue processes for problem solving as described above were based on representations and compliance to constraint targets. Inner and outer environments and their interface specification relied on the description of parameters, their relationships and goals. Eventually, it seemed implausible to fully describe the constraint sets of the entire search space, neither *a priori* nor *in procedere*.

2.4.1 Knowledge-Based Design Systems

Thus, the focus shifted towards the process that organizes constraints – the design process. A separation between the knowledge-base and the control mechanisms was introduced[11]. As before, the knowledge-base compilation required the structuring of the inner and outer environment. But the shift towards the organizing process required the explicit description and structuring of the designing process, its hierarchy and sub-components such as decision-making instances, search behaviours and strategies, iterative transformations, comparative choices and memory. As a consequence, and in reference to Simon's GPS example, not only constraints and goals require pre-processing but also the generative path into an explicit strategy. The descriptions of Simon and Schön led to an implicit structuring of the design process, deduced from the constraints framework. The observer-designer was tacit and not explicitly formulated while knowledge-based design aimed at formalizing the control mechanisms of the observer, seeking automation of the act of designing. As opposed to Alexander's view of the designers' cognitive skills as obstacle[12], knowledge-based design aimed at defining the creativity of designing and thus could be said to heed Schön's findings (1983) of the creative spark happening within the design process in conjunction between different schema (generative metaphor) and iterative reflection (action). The notion of *creative design* was also introduced to distinguish between routine procedures and unique procedures (Akin 1998) - an odd distinction given that unique designs can emerge from creative applications of routine processes or be based on routine procedures such as creative arrangements of rooms.

---

[11] "*The distinguishing features of knowledge-based systems are the separation of knowledge and control and the predominance of symbolic modelling*." (Gero and Maher 1993, p4)
[12] "T*he dilemma is simple. As time goes on the designer gets more and more control over the process of design. But as he does so, his effort to deal with the increasing cognitive burden actually make it harder and harder for the real causal structure of the problem to express itself in this process*." (Alexander 1964, p73)

One of the strategies of knowledge-based design simulation was to generalize a design domain by decomposing it into as many (or few) components as possible that could be transferred to as many design processes and strategies as possible. John Gero (1990) goes into detailed descriptions of possible knowledge bases and constraints, concepts and strategies that are repetitive across design prototypes or cases[13]. The complete set of components with all its relations constitutes a design schema and as a minimum requires the description of the *function*, *structure* and *behaviour* of the design object (Gero 1990).

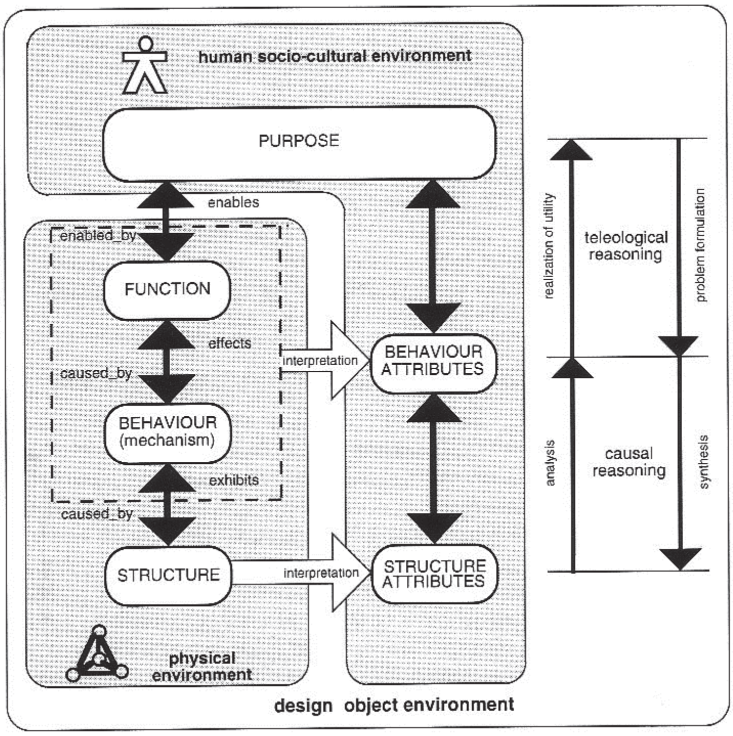


Fig21. *Diagram of a Design Object environment organization and its different types of knowledge required for the knowledge-base (Rosenman el al. 1994)*

The desire to anthropomorphise computing and integrate the knowledge and experience of a designer could be regarded as a positive step. Not only knowledge but modes of creativity were to be encoded through transformation mechanisms of knowledge schema like *combination*, *mutation*, *analogy* or *First Principles* (Rosenman and Gero 1993). The approach was reminiscent of what AI called *case-based reasoning* (CBR) and originally stemmed from psychological research to create knowledge structures called *personal construct* or *preference elicitation* (Kelly 1955). *Decision trees* were constructed from preference choices within schema. Cases describing schema had to be archived as input samples for 'experience' and decisions to connect elements representing associations between sub-categories within schemas. But as Yehuda Kalay (2004) points out, the problem with CBR was first the choice of prototypes and cases and secondly the retrieval of 'good' cases for

[13] "*A design prototype separates function (F), structure (S), expected behaviour (Be) and actual behaviour (Bs). It also stores relational knowledge between them (Kr) as well as qualitative knowledge (Kq), computational knowledge (Kc) and context knowledge (Kct).*" (Gero 1990). Further Gero exemplifies the types of values, variables, parameters, ratios and conditional rules that would constitute the schema 'window'.

the evaluation of states for further transformation. According to Kalay (2004), it is difficult to index cases for retrieval in spatial domains, which deal with non-explicit notions like lines of sight, openness or path-through. Indexing for the retrieval of such discursive cases would lead to many irrelevant cases, complicate 'good' selection and eventually return control back to the designer as searcher. Additionally, the knowledge base is entirely reliant on specific designers who classify good vs bad design cases and thus build subjective knowledge-bases where creativity depends on his preferences. This may be common to traditional subjective design expressions but contradicts the very aim of knowledge-based creative computation to generalize design through automation.

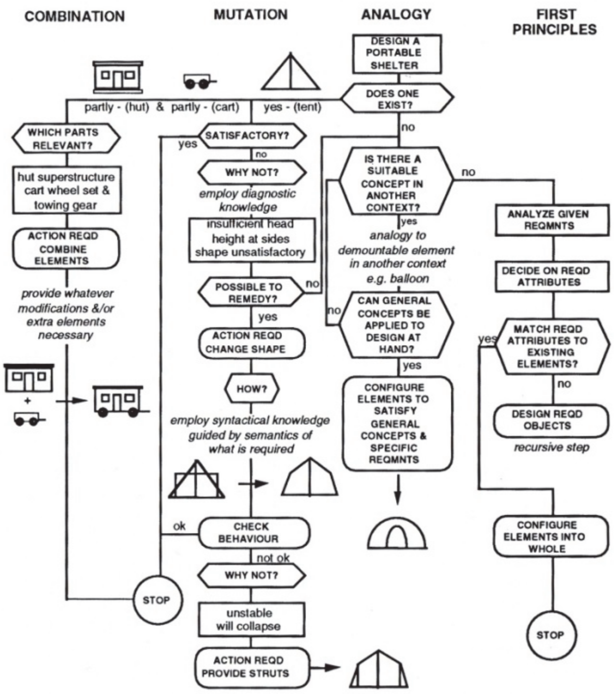


Fig22. *Creative Process in design reasoning (Rosenman and Gero 1993)*

2.4.2 Production Systems

According to Kalay (2004), a trend towards rule-based systems entailed in reaction to CBR and prototype systems. Rule-based systems are based on design rules and rules-of-thumb for producing spatial configurations. Rules-of-thumb, experiential guidelines and rules that do not necessarily guarantee determinable and repetitive results are called *heuristics*. Hence, rule-based systems include heuristic methods such as Omer Akin's Heuristic-based Generation of Layouts (HeGeL) (1992) or Ulrich Flemming's LOOS (Flemming et al 1992) that were called *expert systems*. Expert systems did not attempt to capture exact knowledge content, history and hierarchy but to simulate the design process and therefore the search mechanism for a satisfactory design state. A model's complete set of rules was still coined knowledge-base and thus still belonged to the family of CBR. An advantage of rule-based systems over case-based systems was their flexibility and ability to generalize (Kalay 2004). While rule-based systems generally began by implementing a rudimentary knowledge-base, rules were adaptable and extendable. As opposed to hierarchical or case-based structures, where rules are hard-coded and sequentially affect each other, states could be evaluated and transformed through any one of their rules or criteria independently from the rest of the rule-base, because the rules represented separate actions. Rule-based systems implement an *interference engine* by checking a condition (IF) and responding with a set of potential transformation rules (THEN). This approach was trialled as early as 1969 by Charles Eastman and later called the *left hand side (LHS) – right hand side (RHS)* production system[14] (Eastman 1969). The combination of pre-formed knowledge about the ontology of a design domain of CBR and the heuristic transformation rules of production systems formed the basis for *shape grammars* (Kalay 2004) introduced by Georg Stiny and James Gips in 1972 (1972). As the term grammar indicates production systems were derived from linguistic research and required a semantic base as well as syntactic structure to produce meaningful shapes. The meaningfulness of shapes for the grammars however was usually gleaned from known cases and paired with established design rules, limiting the production system to generate expected outcomes. Creativity rested as before with the designer inventing 'interesting' combinatorial base shapes and still kept the observer-designer in a passive position.

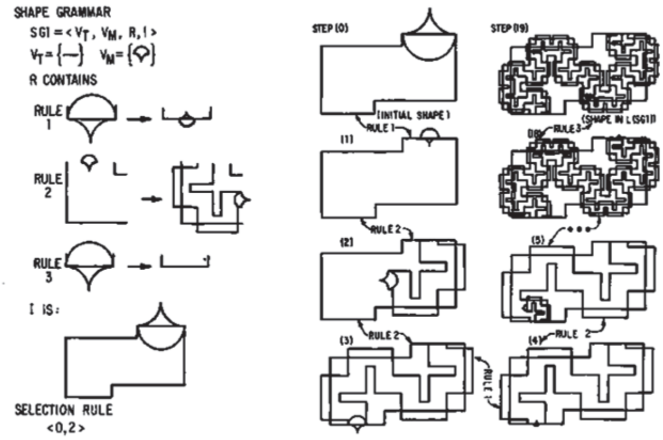


Fig23. *Production System as Shape Grammar (Stiny and Gibs 1972): on left, the rules or grammar of Urform composition with the 'condition -> response' as a model for heuristics of search in design processes (Eastman 1967); right, shows the grammar executed recursively, replacing the condition with the response shape*

2.4.3 Associative Reasoning

Richard Coyne proposed a new direction for knowledge-based systems that would eventually link to a new paradigm of models discussed in the next section. In 1988 Coyne (1988) still described his Spatial Synthesis model with three clearly distinct knowledge functions: form grammar, action grammar and plan grammar. Coyne aimed to embed only *well-behaved* rules, which should not be over-specified for simplicity and hence represent high level abstractions of models of architectural logic. Instead of building large knowledge-bases he started to give each grammar limited authority over the production to distribute control, calling it a *procedural network*. In this network, the grammar of actions represented the most generic part and therefore a *meta-grammar* that could be transferred to many design domains (similar to meta-heuristics in computing science).

---

14 "*By a heuristic is meant a relation between some part of the current problem state and some part of the desirable next state. Most models of heuristics have framed them as productions in a Markov system. The production takes the pattern of Condition -> Response. If the l e f t hand side of the condition is met, then the right hand side is applied to determine or partially determine the next transformation to be made*." (Eastman 1969, p673)

Coyne continued this approach into distributing control not only by encoding observer knowledge but by giving computational models the option to produce knowledge that is outside the observer's control, attempting to pass some degree of autonomy to the model. To this end, Coyne applied the Parallel Distributed Processor model (PDP) of Rumelhart and McClelland (1986) to the interpretation of architectural classes (Coyne and Newton 1990). The PDP belongs to the family of AI learning models called artificial neural networks (ANN) that abstract signal processing in the human brain (Rumelhart and McClelland 1986). The PDP was based on a simple ANN known as *perceptron*, a *feed-forward* network developed as early as 1957 by Frank Rosenblatt (1958), and belongs to the category of supervised networks used to re-construct patterns from association. It does so by adjusting scalars, or weights, between nodes of hierarchically arranged layers. The nodes' values are summed up to arrive at a number that need to match the input pattern. The association is attributed to differences in numerical values between input samples, nodes and output, in the sense that quite literally numerical positions in a vector are compared and adjusted according to some portion of difference (see 3.5). Numerical positions express the intensity of features such as 'height'. The decision to trigger an adjustment depends on threshold functions that decide whether a pattern belongs to a class, i.e. a set of numerical values is too different to be adjusted or not.

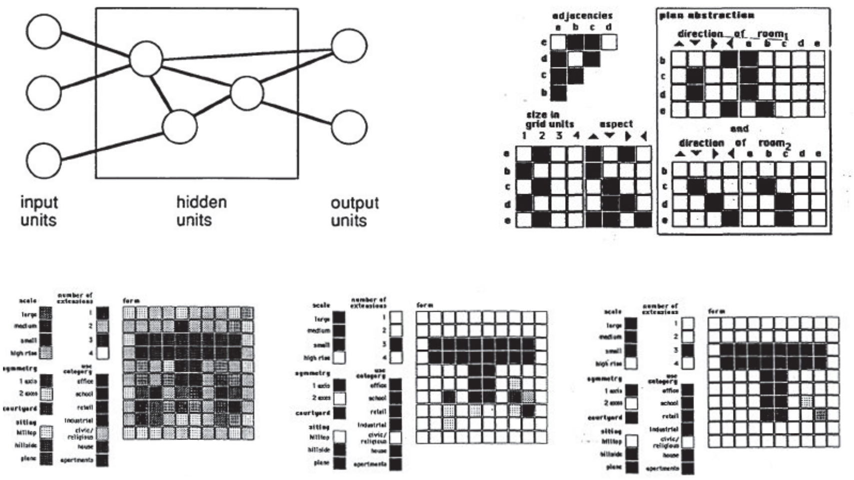


Fig24. *Richard Coyne (Coyne and Postmus 1990): arranged image of (top left) a simplified PDP perceptron; (top right) an adjacency matrix for PDP analysis input of an area schedule and (bottom) three stages in the PDP organization process for a building plan*

Using Rumelhart's model as a template, Coyne is trying to break out of the tautology of having to specify all possible design knowledge bases, mechanism and strategies *a priori*, and to move towards the ability of a model to generate its own representation or rules: "*There is no explicit representation of a schema. However, a schema is implicit in the pattern of associations generated by the system during the learning process*" (Coyne and Newton 1990, p40). Because a schema's ontology is essentially defined by selecting input and target samples by association to experience, Coyne calls the development of classifying spatial descriptions *episodic* or *intuitive*.

On the other hand, a designer does have to specify the features that require classification. Therefore, although Coyne explains how the PDP might be able to distinguish into categories of 'kinds of', 'dependent', 'part of' and 'sibling' (Coyne and Newton 1990), these categories are really already established in the structuring of the input samples. The mechanism for attributing the samples to categories is however not explicit and generally considered a *black box*. Models of this type of constraint-based *associative reasoning* are target-based and are called supervised. While the search mechanism (also called *learning* in ANNs) itself is unknown and controlled by the computational model, the targets are set. Supervised networks lend themselves ideally for causal association to reconstruct patterns as done in pattern recognition.

CONNECTIONISM

Coyne's identification of the PDP's relevance to architectural design constituted a turning point, as it raised doubts about the usefulness of computation as a replication of *designing* by copying or automating knowledge and behaviours of designers. It also attempted to avoid the strong control and direct correlation between constraints specification and design space.

Contemplating computational processes to be semi-autonomous, un-controlled and distributed through a network, puts Coyne into the scientific and cultural zeitgeist of postmodernism (Cilliers 1998) that is reoccurring now. System theoretical approaches of structuring knowledge into symbol systems for manipulating and producing logical expression were abandoned in favour of connectionism. Reminiscent of Simon's *table of connections,* connectionist models propose learning by association between contextual inputs and target outputs that are based however on lower level representations, mainly numerical values that are less structured than symbols, deriving their meaning from networks of distributed data units. Connectionist models were only part of a surge in developments based on networks, population-thinking and social systems that had their roots in language theory, biology and neuroscience. Knowledge and skills of designers and their production processes became accepted as discursive since many of their decisions are taken ad-hoc through intuition and contextual stimuli. Automation of explicit design processes became less desirable whereas implicit design knowledge more valuable. As will be described in the next section, computational models were to become partners of designers within a self-organizing system reminiscent of Schön's reflective conversation.

2.5 EPISTEMIC AUTONOMY OF THE GENERATIVE MODEL

"*This paper rejects the notion that a CAD approach should reflect the traditional non-CAD architectural methodology on the grounds that […] imitating the human process is unlikely in any case to represent the most imaginative use of a machine.*" (Frazer 1995, p60)

Although this second strand of designers and researchers (born around end of WWII) also started out on the basis of cybernetics and systems theory, they did not see the potential of computation in the automation of the design process or the imitation of design reasoning but in the immanent processes and expressions that computation offers (Marshall McLuhan (1994) called technology an extension of human actions, not a replacement). The focus was not to categorize domains of design as there are no general classes of wicked problems according to Rittel[15] but rather the effects of computation on design concepts, models of space, spatial production and occupation. As Rittel and Frazer put it respectively, design is not a scientific problem-solving activity but a concept driven search: "*[Simon's] scientific method does not recognize the need for a generating concept when approaching design, and as a consequence design has come to be misunderstood as a problem-solving activity*" (Frazer 1995, p15). Knowledge has to be generated, not hard-coded. Therefore, the design process and the designer were less of an object of study than the knowledge that could be produced through the availability of computing machines, their processes and design ecologies. Through the application of computational theory new models of space and use should reveal new concepts of design, which in turn promote the computer to become a partner in a collaborative design act of equal immanent strengths.[16]

It could be argued that this parallel strand of researchers was taking computation quite literally as it attempted to work on the basis of its mechanical properties: collections of simple units producing representations of phenomena based on some transformation rules through distributed processing via interaction between reading and writing. The properties inherent in the hardware of computers are reflected in the systemic structures proposed by systems theory, artificial life, some strands of artificial intelligence (like connectionism) and self-organization. The location of control is shifting from a controller-observer (as still present in Coyne's Spatial Synthesis model) to a population of actors-observers (basic units of the system).

2.5.1 Epistemic Autonomy

The exploration of unsupervised knowledge production was already key to Gordon Pask's *Conversation Theory* (1976) and second order cybernetics. Pask in particular provided the foundation to a new design generation approach in architectural computation through his analogue system models like the *Electrochemical Computer* (Cariani 1993), which informed many mostly British architectural cyberneticians like Frazer, Glanville, Coates or Gage. Concepts were introduced that have recently gained momentum like self-organization or emergence. A *solution concept* as proposed by Rittel or Frazer had to be generated by the system itself under guidance through rewards. Rewards in the traditional sense could be the *relevance criteria* for the transformation of a system. But for scientific Cyberneticians like Pask (1958), reward represented the ability to reproduce the system's structure and therefore the maintenance of the system's own logic by which its parts take decisions. The ability to construct *relevance criteria* for rewarding actions depends

---

[15] "*There are no classes of wicked problems in the sense that principles of solution can be developed to fit all members of a class.*" (Rittel and Webber 1973, p164)
[16] "*An architect would not and should not confront a 'criteria machine' to decrease visual privacy, increase public access, and watch contortions of form on a television screen. Instead, in the rhythm of a dialogue, a solution-generating capacity would be an evolutionary enterprise where the machine would act in 'interrupt' or 'reply' to its partners' activity.*" (Negroponte 1970, p39)

on the material properties of a system's parts as communicative relationships between parts could only be modelled through its physical structure.

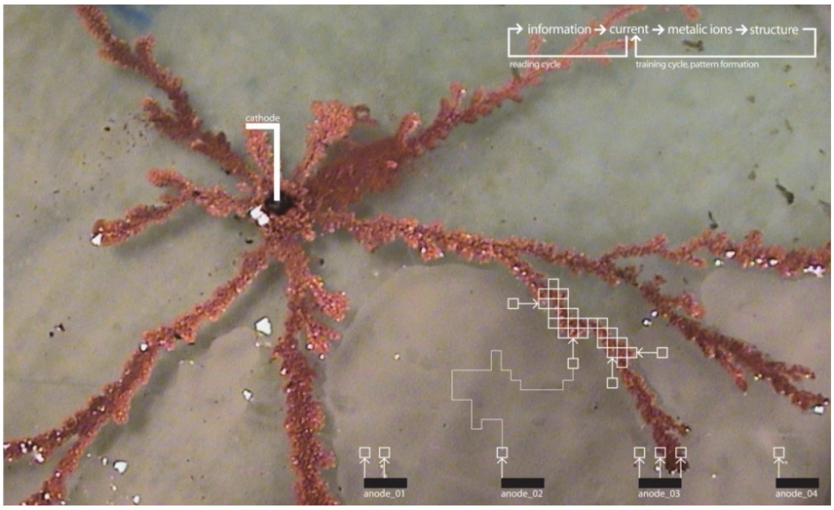


***Fig25**. Pablo Miranda (2006): new prototype of Pask's electro-chemical computer (Cariani 1993) called Dendroid, being able to grow a cubic transformation matrix from an artificial eye; Dendroid served as a demonstration of the epistemology of material computing*

The materiality of systems can find various representations. John Frazer (1995) took the analogue nature of Pask's early models as a template to build material computers such as the Generator project with Cedric Price in 1980 or the Universal Constructor in 1990. Paul Coates (2010) on the other hand interpreted the 'text' of computer code to be the material of generative systems[17]. Both had in common that no matter what materiality the system was embodied through, the local relations between the parts should generate observable emergent concepts of space. Informational openness, a key requirement for the reproduction of a system's structure, was achieved for Frazer via interaction of sensors with their environment for data collection and for Coates via digital interaction between the system internal parts and a database of digitally encoded environments. The design systems therefore achieved *epistemic autonomy* by being informationally open yet operationally closed, meaning the observer-designer did not explicitly direct the search for a solution concept but had to leave it up to the system – analogue, digital or hybrid – to construct meaningful spaces that the observer had to visually and semantically interpret. In other words, to be successful, the observer had to associate an observable state of a model to an experienced schema of his own. The schema and its phenomena are (ideally) not specified anywhere in the system *a priori* but "*an emergent event can be defined as the point where the observer's model breaks down, or in Rosen's terms, the deviation of the observed behaviour from the behaviour predicted by a model*" (Cariani 1993, p27). Knowledge-based design sciences attempted the opposite by re-creating known states from provided schema.

[17] "*When algorithms are expressed as text, in some language, then the distance between the description of the algorithm and the intended outcome becomes greater, and this abstraction into 'real' language (as opposed to the metaphorical 'languages of form' for instance) gives access to the infinite variety of generative grammar.*" (Coates 2010, p2)

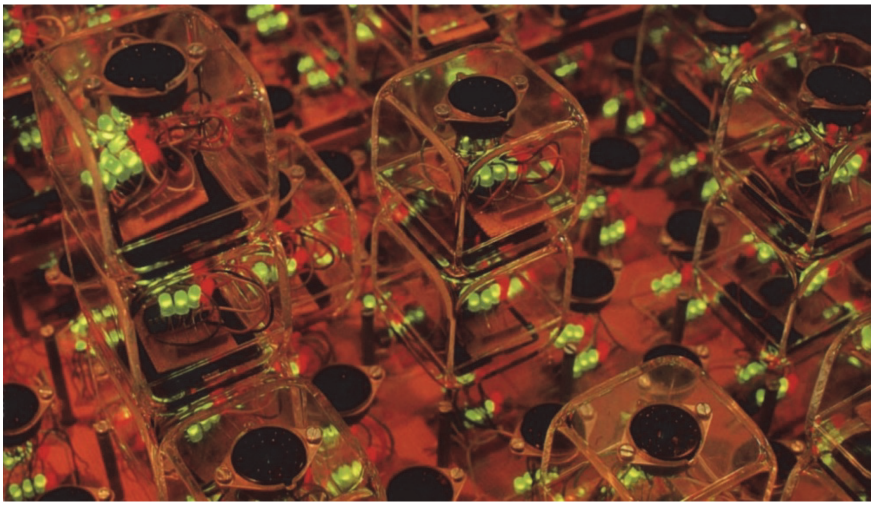


***Fig26**. John Frazer (1995): Universal Constructor with Miles Dobson – the material units equipped with sensors evaluate their context and give feedback on a screen for further configuration by user*

2.5.2 Agency of Situated Actions

Another distinction between knowledge-based scientific design models and models of epistemic autonomy lays in the role and actions of the representation of the parts. In knowledge-based systems, parts can be complicated constructs that increase in size and knowledge over time via a linearly increasing process. The parts of autonomous systems on the other hand contain no knowledge, little intelligence and only limited action rules. They perform their actions repeatedly, might change state but remain structurally the same over time. Data that is processed is gleaned from the environment, which can also be neighbourhoods of parts from the same system. Emergent phenomena for concepts or schema association are produced via simultaneous repetitive processing of large amounts of simple parts rather than many serial transformations on few complex parts. Hence, the emphasis of the system shifts from data configurations to processes over time. The representation of knowledge is located in the relational processes with low abstraction, rather than mathematically complex descriptions incrementally layered. Simon's *generator path* (1969) became abolished and replaced by many simple generators. Emphasis was given to a move from mathematics to algorithms. Mathematics represented the language of the observer, algorithms the language of the actors. Architectural space became to be perceived as a collection of events that is algorithmically defined as a dynamic set of relations between actors, including the designer-observer who must act materially. This autonomous process-based knowledge generation via the new medium of the computer and represented in algorithmic structures will be called the *New Epistemology*, as proposed by Coates (2010)[18]. The New Epistemologists (community of researchers here discussed) is indebted to Jean Piaget's theory of generating knowledge through experiences (constructivism) and Seymour Papert's theory of learning through discovery (constructionism), itself based on Piaget's constructivism (Papert 1980). New Epistemologists regard the designer as a 'discoverer' of spatial phenomena by interacting with the environment through autonomous algorithmic models that encode their assumptions. Therefore, he role of

[18] Coates eventually understood New Epistemology as a new discipline that sat between building sciences and art theory (Coates 2010, p1).

the designer as single author of design became increasingly questioned in line with a post-modern *zeitgeist* of philosophers like Roland Barthes or Henri Lefebvre[19], because it was assumed that space could eventually be simulated as emergent states self-organized by its actors – social, physical or ephemeral.

While the approach apparently distributed control and did away with a distinction between designer, observer and system and as such made it difficult to talk about a structured design system, it fostered valuable novel thinking by forcing the designer-observer to quite literally consider the perspective (or agency) of the simplest and smallest parts involved in the production of space. The role of the designer on the one hand is to view architectural space as situated from within an organism (forging a conceptual link to the early 20th century organic architects like Scharoun) and transfer some actions to its simulated parts; on the other hand as observer, he is to evaluate as a quasi-client agent the successful generation of design states. At times, the situating within the system created quasi-anthropomorphic associations between observer and systems, for example Price and Frazer's intention to create conscious buildings that experience 'boredom' and 'fun' (Frazer 1995, p41).

The approach of epistemic autonomy to design had its roots not only in Cybernetics but also related contemporary fields such as computing science, biology or sociology. All were exploring similar distributed representations with bottom-up self-organizing distributed processes to generate global phenomena. The algorithmic paradigm was expressed through computational models that emulated communication mechanisms from natural and social systems on a simple unit-based level, such as cellular automata, neural networks, evolutionary algorithms, Lindenmayer systems and fractals[20], simulated annealing, agent-based systems and others, which eventually split from the field of artificial intelligence to become known as artificial life (AL) (Langton 1995). By today the list of algorithms and procedures that define generic heuristics of search for ill-defined problems has exploded and is generally called *meta-heuristics* in computing science.

New Epistemology has allowed architects a preview of what might one day be the domains of computational design and the roles of the architect – the *extended architect* as Frazer (1995, p100) calls it as a reference to McLuhan (see above). They created a link between computational models and environment that the original computer artists in the 1960s around Max Bense (Klütsch 2007) with their aleatoric generative procedures neglected. But they would not compromise the orthodox systems thinking for applicability in real design settings where synergetic models and more user-integration are required, such as suggested by Hermann Haken's global-local self-organizing systems called *synergetics* (Haken 1971).

---

[19] Barthes proposed that writing and reading as interpretation are not linearly related and thus an author is not responsible for the meanings instilled in a work. An artefact and its intention draw from cultural and social dynamics that are also dependent on the vehicle of production (Barthes 1977). Lefebvre regarded space as a socio-temporal construct and introduced the *dialectics of space* (Lefebvre 1991).

[20] Evolutionary algorithms, L-systems and fractals (recursive formal systems) work on more complicated composed elements. But the notion of distributed representation and complexity built on simple transformations applies.

2.6 EXPLICIT MODELS OF ENVIRONMENT

"On the one hand, the work requires us to find appropriate mathematical representations which are isomorphic to the spatial and physical form of the building, site or urban area; and on the other the modelling of patterns of activities at these scales." (March, Echenique and Dickens 1971, p275)

John Frazer researched at Cambridge University around 1970 when John Conway was developing the Game of Life model with cellular automata there (Langton 1995), an AL technique later used extensively by exponents of the New Epistemologist. But during that time another institute of architectural science at Cambridge was dominating the British architectural research scene (Keller 2006): the Centre for Land Use and Built Form Studies (LUBFS).

Research at LUBFS was less concerned with issues of artificial intelligence, cybernetics or operations research than with mathematical descriptions of the environment, both in terms of shape and structure, regarding topology and geometric relationships. LUBFS was set up by Leslie Martin who wanted to build on new mathematical methods like set and graph theories or network analysis, introduced to the field of architecture by Christopher Alexander (1964) in his PhD thesis *Notes on the Synthesis of Form*. Alexander was a mathematics undergraduate in Cambridge and a colleague of Lionel March who would help Martin to set up the centre and become the central researcher at the LUBFS (Keller 2006). Alexander's approach called for a rational approach to design with the design process analytically structured *a priori* into a system structure with sub sets of problems and solution targets. His description of the design process were set and graph theoretical observations based on semantic assumptions as proposed in his Appendix I/ A Worked Example (Alexander 1964).

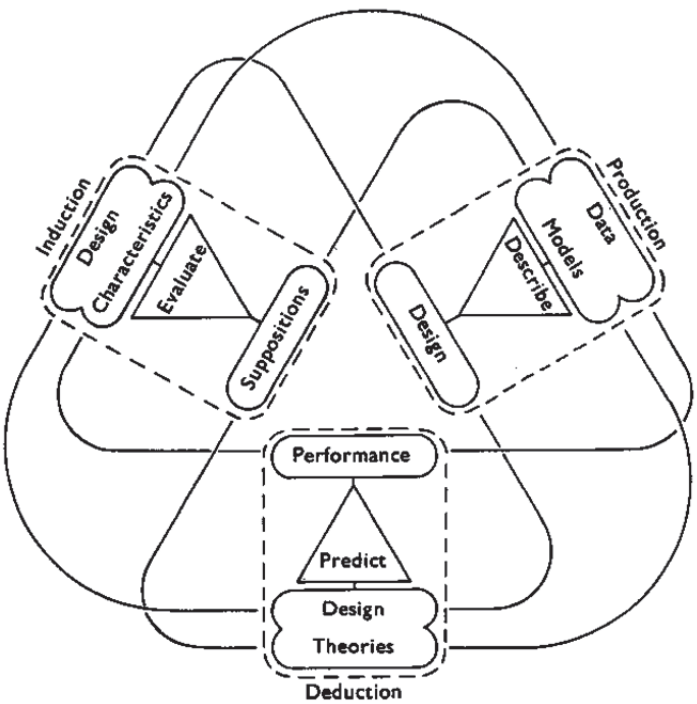


Fig27. *Diagram of a rational design process by Lionel March (Keller 2006)*

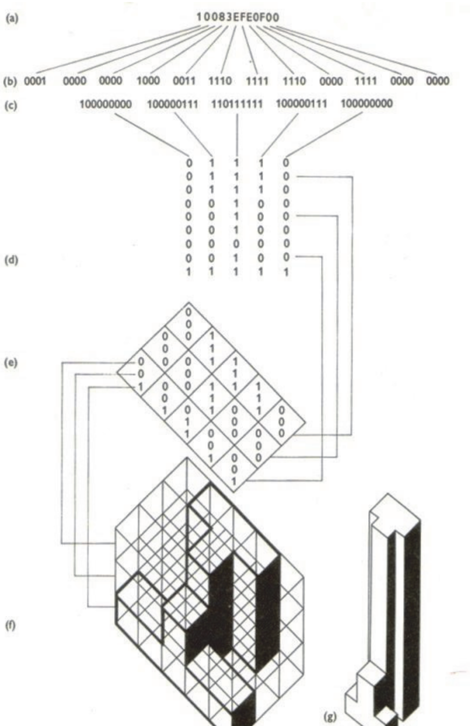


***Fig28**. March and Steadman (1971): encoding Mies van der Rohe's Seagram building mass by Boolean representation of solid/void and encoding the three-dimensional Boolean states via matrices into hexadecimals*

Another influence at the centre was Peter Eisenman's (1963) PhD at Cambridge about formal systems of architecture through graphic transformations based on mathematics. Both Alexander's and Eisenman's work was concerned with what March and Steadman expressed in their preface to Geometry of Environment (1971, p8) as "*the new mathematics appears to be similar to – or isomorphic with – physical and spatial aspects of buildings*". But the centre would extend the systemic and formal exploration by applying mathematical descriptions also to architectural programme and activity patterns in order to represent topological conditions: "*previously geometry was employed to measure properties of space […], whereas the new mathematical theories of sets, groups and graphs – to name but a few – enable us to describe structural relationships which cannot be expressed in metrical forms, for example, 'adjacent to', 'in the neighbourhood of', 'contained by'*" (March and Steadman 1971, p8).

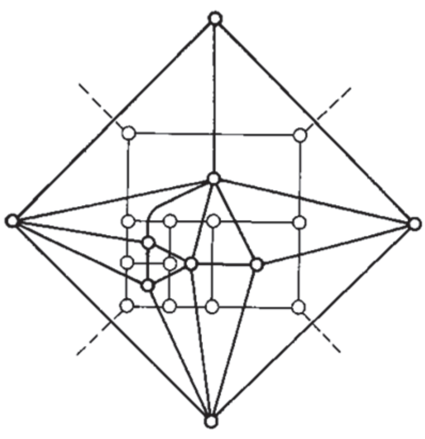


***Fig29**. A dual representation by Philip Steadman of a plan and an adjacency graph from 1973 (Keller 2006)*

Philip Steadman developed Alexander's mathematical methods and semantic assumptions and demonstrated their applicability on a building scale (as opposed to abstract networks mostly on urban scale). Through analogy to utility planning he elaborated graph theories for space allocation by creating dual representations for room layouts and adjacency requirements, applying mathematical simulation or computer programming to architectural programming (Steadman 1970; March and Steadman 1971)[21]. Further, he introduced models from electrical networks for circulation planning with flows, capacities and spatial dimensions. This mathematical approach provided a robust framework for layout classifications, which was later however shown to be limited by the complexity of the accommodation schedule (i.e. exhaustive enumeration runs quickly into problems of operability (Keller 2006) when the complexity of the programme increases due to the exponentially increasing relations between elements).

2.6.1 Mathematical Representations of Building Operations

Steadman (1971, p321), Philip Tabor (1971, p315) and Tom Willoughby (1971, p314) developed new building layout representations that explored geometric configurations integrating circulation performance evaluation based on Steadman's graph representation of geometric relations. This first prototype of generative-analytical integration regarded the adjacency requirements and scheduling between rooms (connectivity) to constitute their *associative* criteria from which operational patterns of organizations could be matched and elaborated (Willoughby, 1971). Tabor extended research into associative networks for organizational structures by looking at classification techniques beyond the matrix to visualize the complexities in the relationships and possible isomorphisms between space and operational organization (Tabor, 1971). He employed diagrams, or *maps* as he called them, that can be automated from numerical data such as dendrograms or Venn diagrams that add another layer of visualization of organizational structures on top of graphs or matrices.

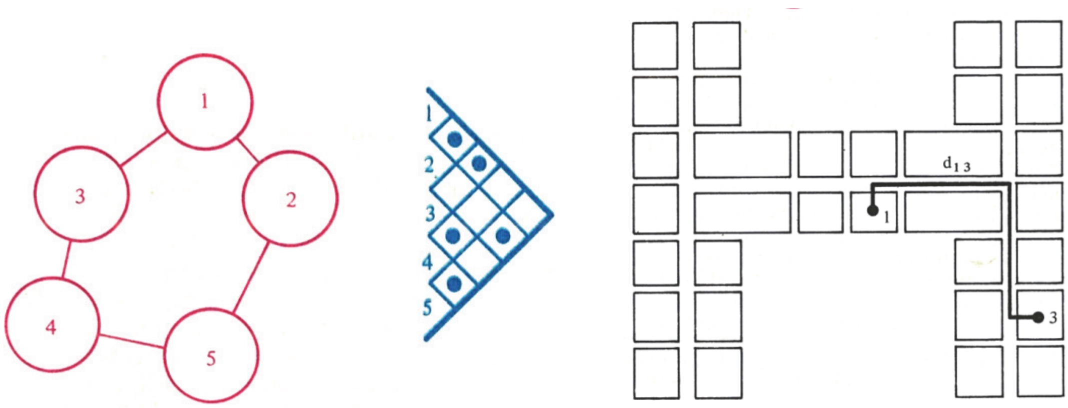


Fig30*.Three representations of room adjacency and movement route by Tom Willoughby (1971)*

The work of Steadman, Tabor and Willoughby built on Alexander's mathematical models of environment but added new dimensions of occupation, activity and programmatic associations. They acknowledged that design methods purely reliant on mathematical calculations and scientific models were limited to some functional domains and tested the limits of operability and complexity of exhaustive

[21] *Programming Programming* as Keller (2006, p52) would call it

enumeration models for design spaces by computer programming (Steadman 2014, p27. This empirical insight from hands-on research made them critical of the rationalist paradigm by Alexander, who proposed that the design process could be entirely automated through scientific models that require total quantification. In 1976 Lionel March stated that "*A scientific hypothesis is not the same thing as a design hypothesis. A logical proposal is not to be mistaken for a design proposal*" (March 1976, p15). While he held on to the vision of 'simulated environments' that encompass all possible stakeholder descriptions (Keller 2006), he came to describe the design process as an evolutionary process ('evolutionary history'), meaning serial or iterative procedures guided by the designer but not determined by a designer's *a priori* intention, similar to Schön's *conversations* or Negroponte's *dialogues*.

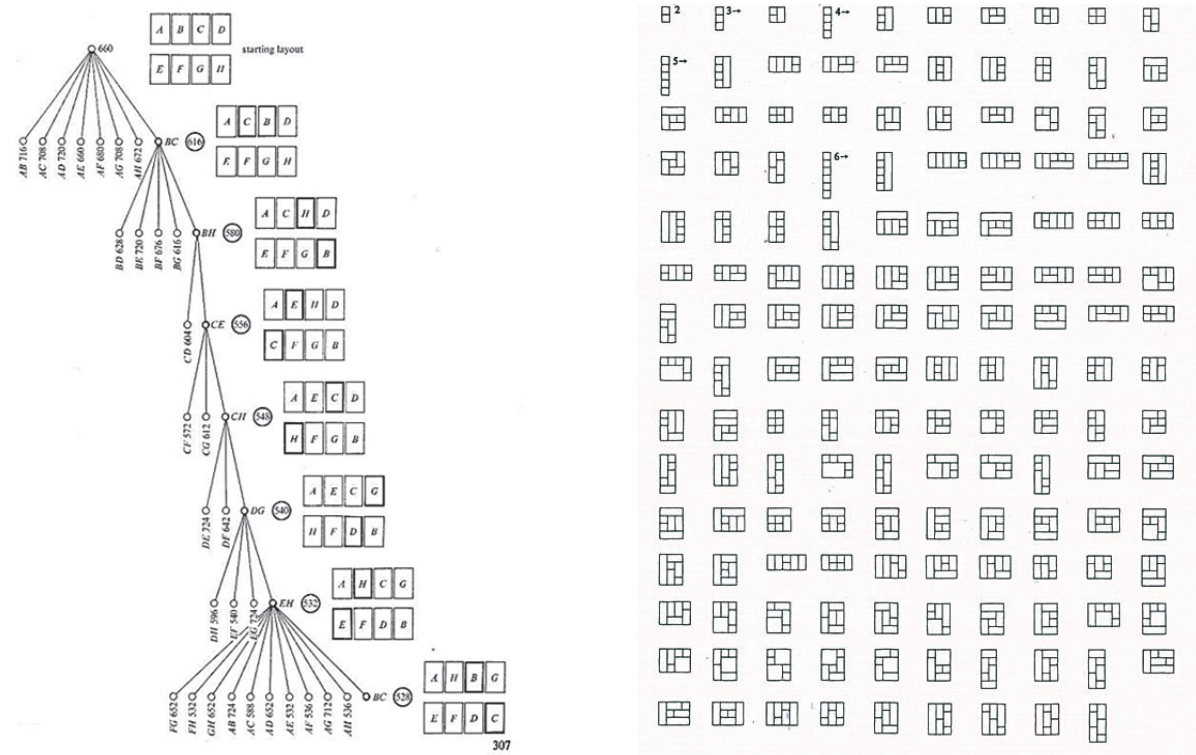


Fig31. *Room activity allocation method (left) using label swapping to generate permutations of possible adjacencies that comply with constrained adjacencies (March and Steadman 1971); (right) exhaustive enumeration of possible dimensionless 'rectangular dissections' of =< six rooms (Mitchell, Steadman and Liggett 1976)*

2.6.2 Positive Limits of Mathematical Approach

The limitations outlined by members of LUBFS who explored exhaustive enumeration and automatic programme allocation should be regarded as a benefit to later generations of architects using computation for at least two reasons:
First, the understanding that architectural design spaces even if limited to concise aspects are very complex and require distinct isomorphic representations. Despite the massive increase in computational processing power since then, there has been no example of an architectural design problem that can be generalized and automated within a domain when an association to more than one performance or simulated activity is attempted. Experience across the design sector – academic and professional - confirms that disaggregating simulation into parsimonious models

enables the integration of non-quantifiable knowledge and therefore the designer[22]. This limitation also allows for some transfer of application to other design briefs within the same domain by being transparent and simple enough for adaptation. And secondly, the individual representations and models developed at the LUBFS provided elementary components of architectural representation as much as the computing methods of the New Epistemologist provided elemental processing models. Had they been absorbed or developed into larger more opaque simulations their clarity and value to design representations for 'environments of man' might have not been so accessible.

The LUBFS can be accredited with having developed many geometric representations later built upon like Steadman's graph representations[23] or Tabor's diagrammatic visualizations of organizational classification. But equally, Keller would argue, the very intention of wanting to create objective quantitative representations of architecture that aimed at spatial structure beyond appearances – topology over geometry – unintentionally paved the way for the formalism of the next thirty years in architectural computing, because design geometry could be parametricized and shape therefore easily transformed while maintaining its topological structure. Charles Jencks (1971) in fact labelled the predominant architectural paradigm at the beginning of the 70s as Parametric in his chart Towards the Year 2000. A paradigm enduring to this day producing an aesthetic often divorced from the architectural programme or activity.

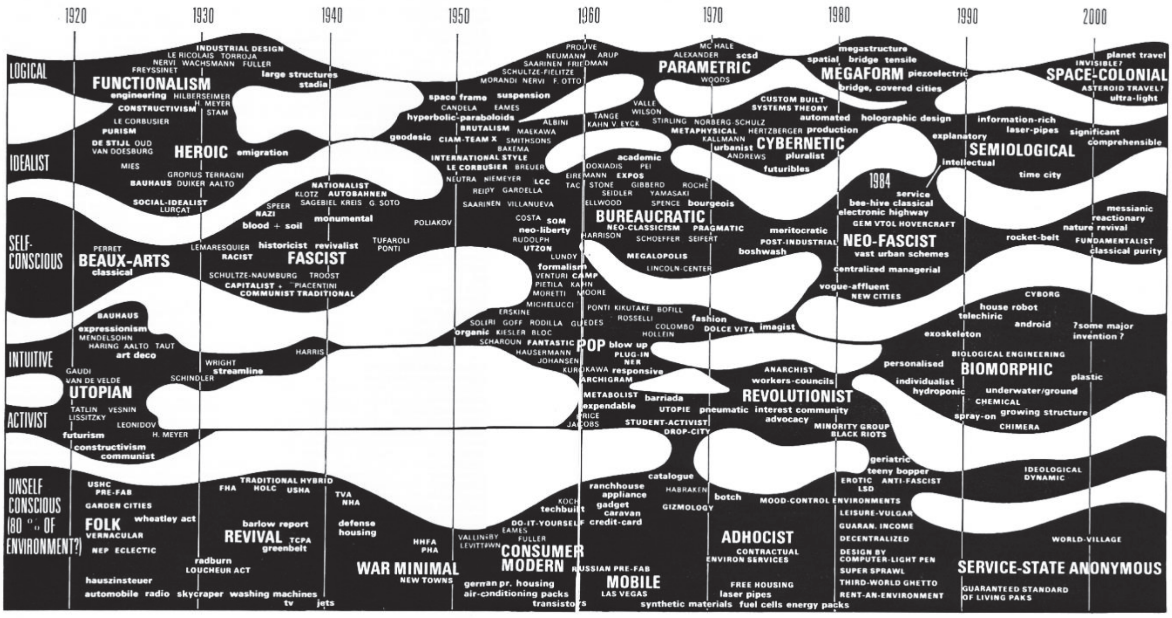


***Fig32**. Charles Jencks (1971) diagram Towards the Year 2000, defining the work around of the 1970s perioid as 'Parametric'*

---

[22] Liggett concludes in her review of automated facilities layout: "*In spite of the long research history associated with automated layout and space allocation systems, in practice these systems have not been utilized to their full potential.*" (Liggett 2000, p213)

[23] Steadman's dual graph for movement structures like circulation became foundational for some Space Syntax techniques like the Axial Map which uses a dual graph representation to analyse (street) network integration according to lines of sight/ unobstructed movement in a map and a connectivity graph for centrality (Hillier and Hanson 1984; Crucitti, Latora and Porta 2006).

2.7 BEHAVIOURAL CONFIGURATIONS OF SPACE

"(but) the relation between space and social existence does not lie at the level of the individual space, or individual activity. It lies in the relations between configurations of people and configurations of space." (Hillier 1996, p20)

While the New Epistemologists search for expressions of space inherent in algorithmic systems and Steadman and colleagues created mathematical representations for geometric permutations of explicit space, another type of representation was proposed in the 1970s: patterns of occupation mapping spatial configurations. The basic assumption was that the configuration of a space or environment contains patterns of behaviour that generate it. Hence, there must be a mutual mapping between occupational and spatial representation.

In How is Design Possible, Bill Hillier and Adrian Leaman (1974) proposed that culture could be regarded as an artificial system which contains instructions for behaviours that we learn: that the built environment is a reflection of the unfolding of un-consciously learned behaviours, reminiscent of Chomsky's linguistic rule-structures that are used implicitly. Also designers follow learned procedures that are expressions of an 'unfolding' of cultural constructs, which Hillier and Leaman called *pre-structures* and proposed that cultural constructs can be observed in patterns of occupation. Pre-structures decode a genotype of cultural construct to a contextualized phenotype and can be regarded as the designer's set of heuristics, meaning the design procedures adopted by a designer as a mix of regulatory design constraints and personal design rules. The genotype, or g-model, cannot be directly transformed by an individual designer as it represents a contemporary spatial reflection of social norms and hence could be regarded as an autonomous unconscious set of instructions from which the designer learns his heuristics for generating a design as well as using it as a benchmark to interpret the outcome.[24]

2.7.1 Occupation Behaviour

There are two types of behaviours that affect each other in Hillier and Leaman's proposal (1974): *designing* - the behaviour of the designer who applies learned heuristics stemming from a conventional set of instructions (in the sense of social consensus); and *occupying* - the behaviours of users who produce occupation patterns stemming from a cultural context manifest in the built environment. Learned set of instructions, similar to intuition, were called pre-structures that situate the g-model in context to produce a phenotype or p-model (Hillier and Leaman 1974, p4). Hillier later called occupational patterns the *generic function* of built space (Hillier 1996, p223), which are autonomous from deterministic or conscious design intervention, because they are embedded in the social g-model. *Generic function* thus means general patterns of presence and co-presence that

[24] Hillier is very careful to point out that the genotype is not meant to be a literal analogy to the biological evolutionary genotype, where direct genetic cellular transformations evolve a specification through contact. Instead, the present genotype is to be understood as spatio-temporal reality, meaning a true presence of space expressed in its built environment. In The Social Logic of Space, the genotype is also called the *inverted genotype*, as the real built environment with all its phenotypes, i.e. individual spatial instances, together form the genotype (Hillier and Hanson 1984).

occur in built spaces independent from design intentions or building brief that are inherent in configurations of space.

Behaviours constituting the *generic function* are activities, which are not specified by functional requirements. Eventually, the set of activities is reduced to movement behaviours, because movement maps spatial configuration most concisely (see section quote). To move from one space to another, or the inhibition to move, is determined by the permeability between spaces. Hence, the permeability structure, or p-complex, is the most elementary expression of occupation that implies the g-model. The configuration of spaces into buildings (or places) embodies p-complexes facilitating types of movement and contains certain *generic functions*. It could be argued that buildings therefore can be classified by their *generic functioning*, i.e. how people cognitively decode a space and behave in or use it.

2.7.2 Designer Behaviour

The second behaviour affected by occupying is *designing*. For Hillier and Leaman (1974), designing constitutes a very constrained activity implementing *pre-structures* evaluating p-models against a g-model so that certain conventions are achieved[25]. Hillier provided an example in Space is the Machine of a *barring process* (1996, p304ff), constituting a step-by-step generative process, albeit a simple case study, of inserting partitions in a grid (like walls) to allow or inhibit movement, thus creating permeability structures. Each step or *design move*, as he calls it, is locally executed but affects the global permeability structure and therefore the *generic function* of the configuration. The global structure in return gives properties to local spaces and enables certain types of occupation. Properties of local spaces are dependent on the p-complex and the degree by which a local space is integrated into the global permeability structure. Integration is measured by topological depth, indicating the connectivity between spaces, and their centrality within the permeability structure, indicating the intrinsic use of a location (i.e., how often the location is travelled through when all routes between all location are considered).

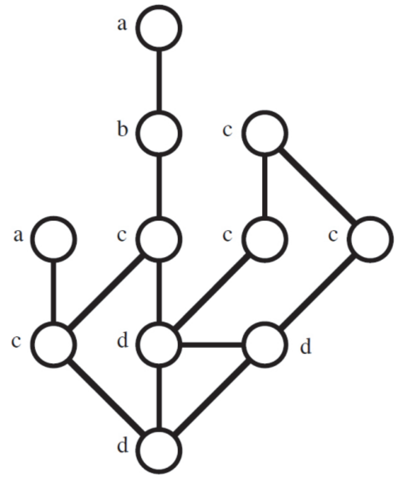


Fig33. *Global effects of the topology of a p-complex, identifying four spatial types: a-d spaces (nodes) dependent on the number of their connections and amount of through movement (Hillier 1996, p249)*

[25] Hillier describes a hierarchy of constraints as three filters that guide the design process as an applied succession (Hillier 1996, p330): 1 generic function, 2 cultural intent (g-model), 3 individual building differences (specific building functions or brief)

Hillier (1996, p251-252) specifies the moves of the *barring process* to have certain types of global effects on integration and he also identifies four types of local spatial properties deriving from the global effects of p-complexes, which he calls a- to d-spaces[26]. Functionally specified, enclosed end-spaces are labelled as *a-spaces*, which do not allow through-movement but only "*to and from themselves*". Transition spaces linking destinations are *b-spaces* that "*cannot in themselves be dead end spaces, but must be on the way to (and back from) at least one dead end space*". Spaces that 'link links' and offer choices of directions are *c-spaces* that "*must lie on a single ring so that cutting a link to a c-type space will automatically reduce the ring to one or more tree (graphs)*". Finally, *d-spaces* are highly resilient movement location because they "*contain at least two rings which have at least one space in common*" providing high choice of movement (Hillier 1996, p251-252). Internal performances of *a-spaces* depend little on the global p-complex as their occupation is spatially regulated by furniture grids and inventory specifications. Only when accessed do they become an essential part of the global movement structure. Design behaviours should therefore focus on the configuration of higher integrated spaces or movement structures that produce global spatial effects and give rise to *generic functions*. Those effects and the *generic function* lend intelligibility to the building that the user unconsciously interprets for occupation and allows him to associate spaces and buildings with expected performances and phenomena.

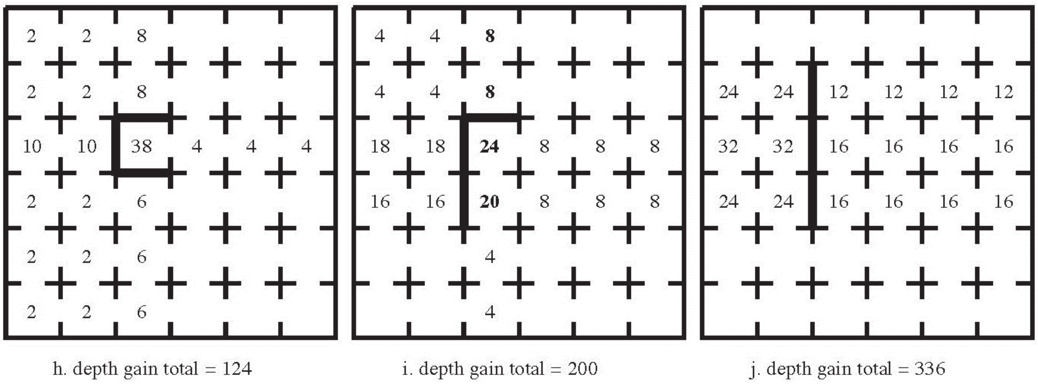


Fig34. *Three alternative states of 'local moves' of Hillier's barring process (1996) showing that each local move constrains consecutive moves by changing the global performance of the configuration (emergence-convergence)*

Hillier argues that types of spatial relations embodying *generic function* emerge from design behaviour or strategy (set of local moves). Similar strategies converge onto similar types of spatial configurations, containing performances and phenomena. With this concept of *emergence-convergence* (1996, 312), Hillier implicitly suggests that classifications of building typologies might be dependent on occupation patterns, movement structures and *generic function*. They are not a reflection of sector specifications for an explicit building use or typology. Buildings could therefore be classified according to p-complexes rather than current ideas of sectors or use typologies (Derix and Jagannath, 2014a).

The relationship between local moves and global effects Hillier calls *local-to-global laws* (1996, p256ff), which in the *barring process* had been associated to a selected

[26] "*Occupation (here specific functional use) uses the local properties of specific spaces, movement the more global properties of the patterns of spaces.*" (Hillier 1996, p248)

few moves and effects that give rise to four permeability properties. Hillier wanted to demonstrate that the designer is not only unconsciously constrained by his *pre-structures* but also the tolerances of the design moves. There are not many moves available to the designer as the effects and spatial constraints of the g-model will limit his freedom. While there are higher tolerances to the initial design moves, each move reduces the options as the emerging *generic function* will govern the remaining moves. *Designing* in this sense represents the search for the maximum amount of moves that comply with a *generic function* and give rise to the highest integration value of the p-complex. This could to be done by carefully arranging a-spaces to achieve desired integration and centrality values within the p-complexes.

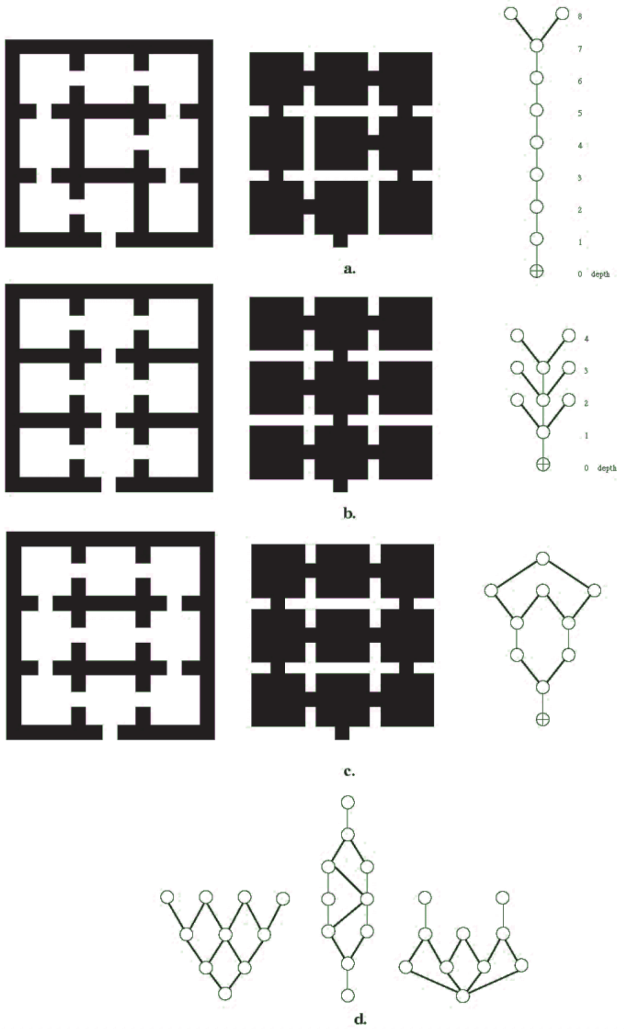


Fig35. *Three area plans (a-complexes), their negative permeability map (p-complex) and their depth graph from their entrance locations; (bottom) three justified graphs representing the same p-complex from different spaces as root node (Hillier 1996)*

2.7.3 Mapping structure between behaviours

It might be apparent by now that behaviours – *occupying* and *designing* - and spatial configurations – designed or unplanned – are assumed to constitute mutual representations: occupational behaviours map spatial configurations that in turn map social genotypes of space, giving rise to *pre-structures* or design behaviours. Originally, Hillier and Leaman (1974) called this cross-referencing a *commutative square* and its mechanism to instantiate the dual structure into reality, a *manifold*. The manifold consists of the designer's code – his heuristics built on *pre-structures* –

and the representational device he uses. As already discussed, *pre-structures* are autonomous from design intentions and consequently so are their representation devices: *"The understanding of all such systems lies in discovering how the internal autonomic structure of the 'simplest structures' of the morphology already contains the rules which govern aggregation into higher logical forms"* (Hillier and Leaman 1974, p3). Rules for aggregation are learned by the designer and built into representational devices that can be any machine or: "*An algorithm is similar. The algorithm constructs its permissible universe, performs a structured conversion on what is 'selected' within the domain so constructed, and outputs an 'interpretation'"* (Hillier and Leaman 1974, p7).

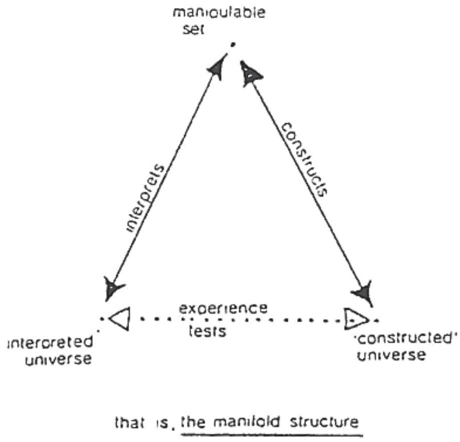


Fig36. *The manifold structure mapping pre-structures and design states into each other (Hillier and Leaman 1974)*

In the search for an alternative theory of design to Alexander's system theoretical approach[27] they anticipated the potential for a *designerly* theory of the built environment to reside in an explicit description of the rules for aggregations and mapping structures that transmit the g-model through *pre-structures*. Eventually in 1976, a morphic language was proposed to map spatial configurations through non-discursive technique (Hillier 1996, p.65ff). With *non-discursive* Hillier intended the externalization and generalization of properties of spatial configurations into analytical method that underlie pre-structures (Hillier 1996, p35). A morphic language borrows elements from mathematics – set theory and logic - and natural language – syntax and semantics - and was represented through graphic elements into an ideographic system, called *Space Syntax* (Hillier et al. 1976). The syntax tries to reduce the entire permissible set of spatial configurations[28] in a generic carrier space to some symbolic operations that implement rules for the aggregation of spatial units. The spatial units represent permeability structures limited to *continuous* and *discontinuous* character. The syntax itself borrows its symbols from mathematical set theory and consist of only one symbol for the relation of two units, namely ⊂ for *containing*, and three types of brackets for the description of

[27] "*We have 'sciences of the artificial' to enable us to 'understand' what we already 'know'.*" (Hillier and Leaman 1974, p2)

[28] One of the key drivers of Space Syntax was the question of redundancy. As with language there are only a few meaningful grammatical constructions that make semantic sense, i.e. not all grammatically correct sentences are semantically permissible (Hillier et al. 1976, p151)

betweeness of units, meaning the alignment or placement between spaces. The key to producing a variety of eight types of operations capable of giving rise to many different spatial aggregates lies in nesting those relations and *betweeness* types. Nesting, or recursion, of containment and alignment operations can express rules that either allow for distributed or controlled (non-distributed) growth. Distributed growth reflects local-to-global rules and controlled growth reflects global-to-local rules. The resulting spatial patterns produced from the aggregate of the two types of spatial units generate local elementary relations representative of the global structure. These local elementary relations are distinguished into *path objects* and *space objects*, which stand for movement and occupation spaces (see Fig37 for the eight syntaxes described by the original paper).

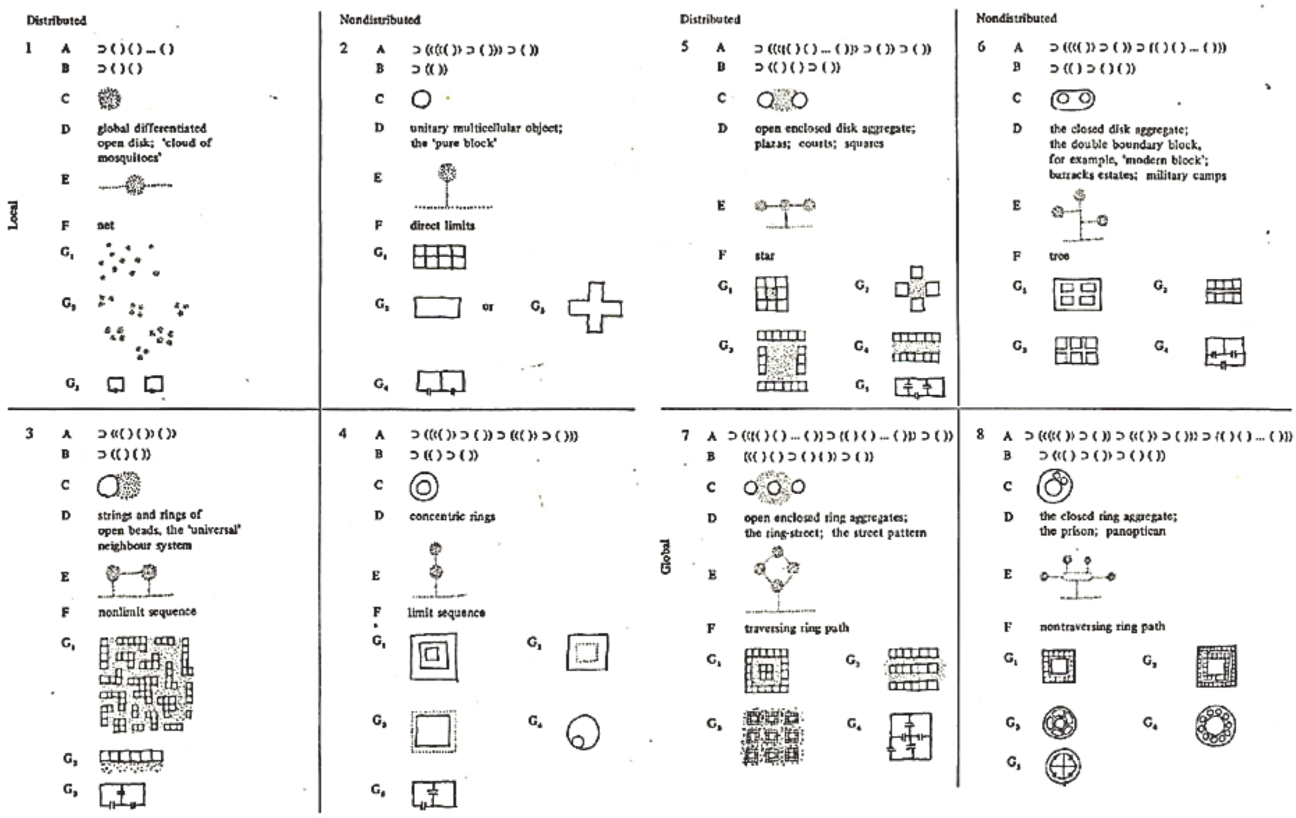


Fig37. *Space Syntax ideographic system (Hillier et al. 1976, p176-177: eight syntaxes generating either distributed or non-distributed permeability pattern. Syntaxes produce from top left (no. 1) local-global structures (local relations govern global patterns of use) to bottom right (no. 8) global-local structures (global relations govern local patterns of use)*

These syntaxes were not meant to represent sets of instructions to support design decisions but to reflect on the generative laws that produce *generic functions* and emergent spatial configurations immanent in social conventions of the built environment (and therefore in designers' *pre-structures*). The above mentioned dual representation of occupational and spatial patterns was described as *inverse law* (Hillier et al. 1976, p179), as people produce spaces and spaces produce occupation that are duals of each other in a configurational sense (Coates and Derix 2007).

2.7.4 Spatial Unit as Relations

Having discussed the autonomy of *pre-structures* and the generative laws from detailed design intentions, another issue important for this thesis was described in the *Space Syntax* paper: an element in a (spatial) system must be understood as the local elementary relations identified in the syntaxes. These relations emerge from "*families of local moves which by following different rules produce spatial effects in the complex*" (Hillier et al. 1976, p257), namely the *generic function*. Spatial units therefore are "*elementary syntactical objects*" (Hillier el al. 1976, p163), not geometric representations of shape. It was believed that the associative syntax – associative as it is not only a generative law but also encodes the g-model that provides interpretation for p-models – replaces the 'knowledge unit' propagated by knowledge-based design. In Space is the Machine, Hillier (1996, p322ff) in fact sees designing as a *conjecture testing* mechanism. That is testing the conjecturing mechanism until its outputs (p-models) can be interpreted correctly. The job of the designer therefore is the reverse mapping of the p-model against a g-model which as a consequence led the Space Syntax community to concentrate on the research of analytical capacities rather than syntactical strategies.

To this end, permeability was further investigated through cognitive properties of space that support movement behaviours. Visibility dependent representations were introduced that give further dimensions to permeability. First a sight-line based graph representation called the *axial map* was introduced in the Social Logic of Space (Hillier and Hanson 1984). In Space is the Machine the use of an even more specific perception-based representation called the *isovist* was introduced. The analytical isovist model was developed by Michael Benedikt (1979) to analyse view sheds and co-presence geometrically. Both measures and representations – the axial map being a topological graph representation, the isovist a grid based gradient map (see 3.4) – enhanced the understanding of choice behaviours by occupants and the description of local spatial phenomena in relation to the global structure.

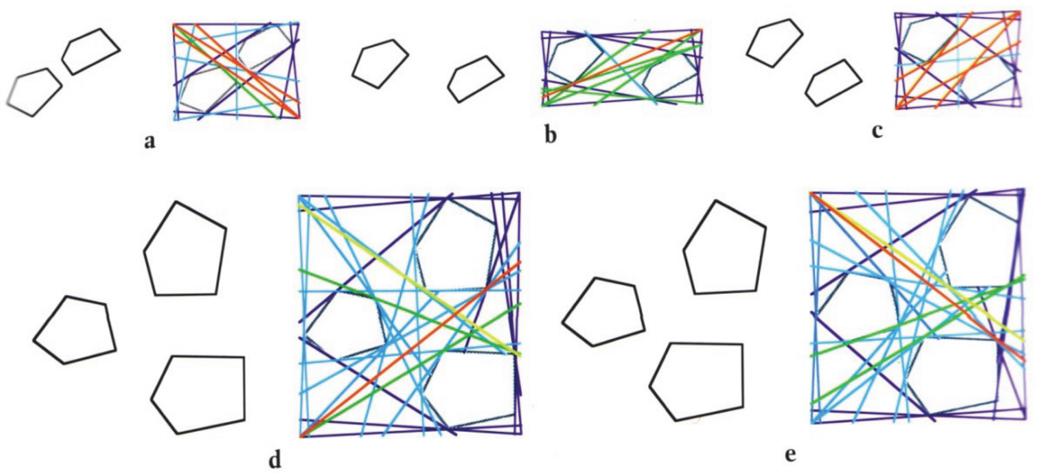


Fig38. *Axial line maps showing highlighting in red the most integrated sightline for various arrangements (Hillier 1996)*

Finally, Hillier and Hanson (1984) produced the *justified graph* representation (j-graphs) for a localized understanding of building depth and spatial connectivity. J-graphs also provide insights to the observer about potential local user choices and more importantly an objective technique for quasi-subjective representation of spatial configurations that differentiates intelligibility into individual locations.
The developments of cognitive representations for spatial configurations based on permeability greatly improved the understanding of how to approach *generic*

function as initially intended by Hillier: *"Generic function refers […] to aspects of human occupancy of buildings that are prior to any of these: to occupy space means to be aware of the relationships of space to others, that to occupy a building means to move about in it, and to move about in a building depends on being able to retain an intelligible picture of it."* (Hillier 1996, p284)

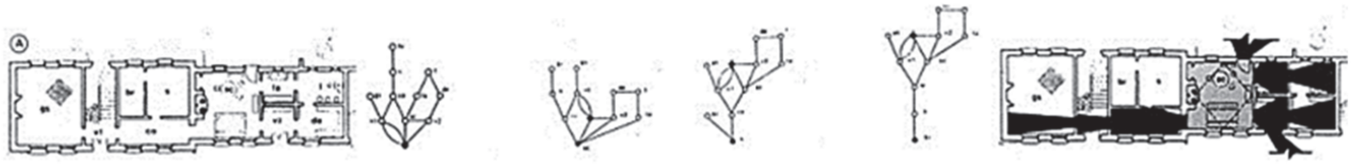


Fig39. *Interpretation of a residential plan (left) through four justified graphs (middle) and isovists (right) (Hillier 1996)*

The original aspiration of Hillier and colleagues for a theory of space was initially a self-organizing generative design theory and progressed towards an exclusively analytical approach. In the original Space Syntax paper a clear position towards a generative bottom-up *designing* system is taken: "*This is what we believe, the minimum syntactical rule for a spatially coherent aggregate, and its global result is that, through the distributed repetition of its local rule it defines the carrier space.*" (Hillier et al. 1976, p164). 20 years later this position had changed to: "*These local to global laws are independent of human volition, and as such must be regarded as more akin to natural laws than contingent matters of human existence*" (Hillier, 1996, p258). Hillier did not intend to abandon design as a generative methodology but the latter position paved the way for the academic community of Space Syntax at UCL to pursue the understanding of immanent natural laws of space, neglecting the *pre-structures* and design heuristics that map genotypes into phenotypes from which spatial configurations emerge.

2.8 CONCLUSIONS

The changes of relationship and roles between the computer as design medium and designer have been discussed in this chapter. The shift of roles from instructing to participating by the designer went hand-in-hand with the shift in purpose for the use of computation from the simulation of design to simulation of space. The selected strands of research and development peaked around the year 2000 when rapid technological advances, i.e. visual programming, stimulated a new generation of users to apply and refine these research strands, albeit still in isolated silos.
A synthesis of the strands discussed above has not been attempted bar a few exceptions such as the professional case studies by CDR and few university courses[29], both of which will provide the basis for this dissertation.

[29] Paul Coates and the author's MSc Computing & Design at UEL between 2000-9; Alasdair Turner's MSc Adaptive Architecture and Computation at UCL between 2004-10; the author's visiting professorship at TU Munich in 2011-12; the the InfAR Institute at the Bauhaus University Weimar since 2013 or the Information Architecture Chair at ETH Zurich since 2013.

3 BASIC ALGORITHMS OF COMPUTATIONAL DESIGN

This chapter provides a brief description of the most commonly used algorithmic models from computing science discussed in the case studies of chapters 4-8. The reader will occasionally be referred back to this chapter for technical descriptions of basic processes and representations. Algorithms discussed in the case studies and not listed here represent derivatives from the described fundamental models and are discussed within the context of the case study where they occur. Because the thesis is not aiming to be technical but conceptual design research, the descriptions are not exhaustive but provide the minimal information necessary for the reader to grasp the principles.

The selection of basic algorithms discussed below mostly stem from the author's nine year teaching curriculum of the MSc Computing & Design at UEL. While new algorithms have been added during the ten years of CDR (3.1.1 and 3.4), the fundamentals are consistent with the CECA models. This is primarily due to the research of the author and CECA focussing on local distributed self-organizing models of spatial organization. Most of the below discussed models belong to the class of *meta-heuristic* algorithms, which mostly display self-organization behaviours and distributed structures. Meta-heuristics represent *search* and *optimization* heuristics that have been generalization into fixed ontologies and algorithms that can be applied to many contexts without changing their core procedures (as opposed to heuristics, which are not generalized and heavily context dependent). Meta-heuristics *satisfice* problems without necessarily finding optimal solution states.

The order of technical descriptions is aligned with their appearance in the case studies.

3.1 GENETIC ALGORITHMS

While different evolutionary algorithms were used at CECA such as the basic genetic algorithm (GA) and genetic programming (GP), the author aimed to focus on GAs and explore more complex variants with students and colleagues of CDR. Thus, multi-criteria optimization with Pareto fronts, steady-state optimization or mixed selection models were explored. The basic GA is based on the pseudo-code of David Goldberg's (1989) algorithm.

3.1.1 Embryology

Genetic algorithms form part of search meta-heuristics. They work on the basis of populations of individuals that evolve over generations via the Darwinian principle of selection of the fittest. Each generation tries to select the best individuals for reproduction to form the next generation. Each individual is based on configurational constraints encoded in an embryology, which determines the metric and topological relations between parameters and thus distils the ontology of a configuration. The embryology also guides the decoding of each individual into a determined representation, for example the ontology of a window, house or urban network. Decoding occurs at the end of the evolutionary process to visualize results.

In most cases the embryology is encoded through binary strings, meaning an array of binary bits that are grouped to represent 'genes'. The collection of several genes produces a chromosome, representing an individual (Derix 2008).

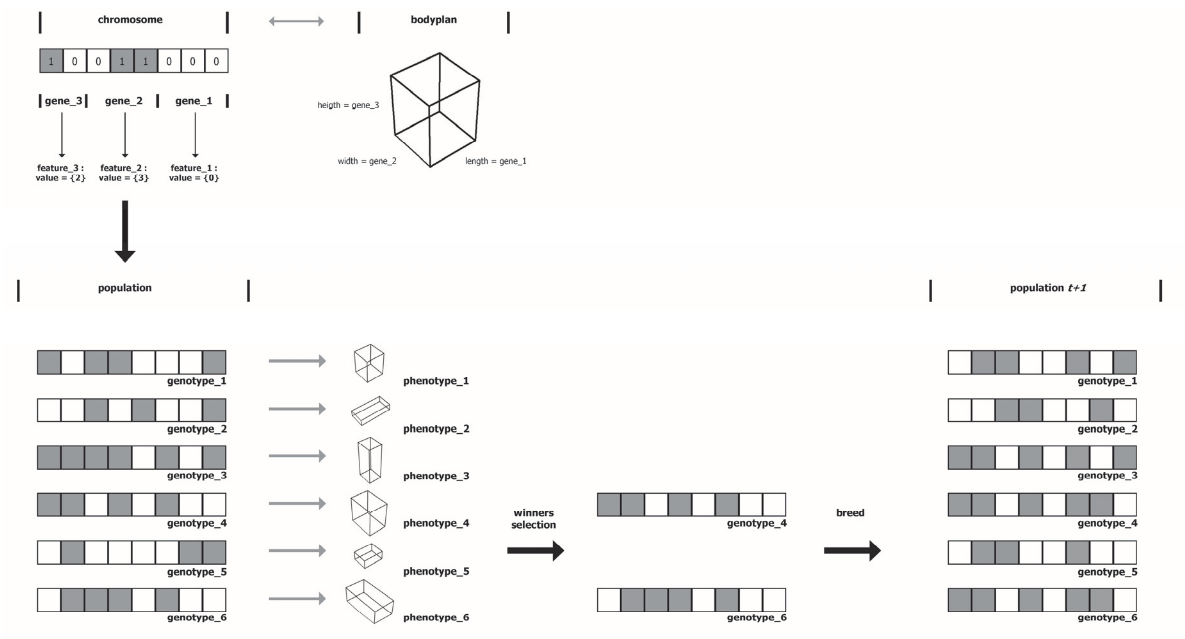


Fig40. *The construction of the embryology with (top) the definition of a simple cubic geometry into a chromosome with three genes; and (below) six 'phenotypes' decoded into their morphology (Derix 2008)*

3.1.2 Fitness

To select the best individuals of a generation for reproduction, the performance of all individuals has to be calculated, called the *fitness*. The fitness can be calculated in many ways but importantly the fitness criteria need to be comparable. A function calculates the fitness for each individual for chosen criteria and normalizes the values into the fitness sum. In the simplest case, only one criterion is identified and no explicit fitness function exists. When more than two criteria need comparing, more complex fitness functions must be applied to reflect trade-offs between heterogeneous dimensions (see below: Pareto fronts).

3.1.3 Selection

Generally, two selection paradigms exist: *natural* and *artificial* selection. Natural selection means the automatic selection by the algorithm based on the encoded selection procedure, while artificial selection means the choosing of individuals by the observer directly without a selection procedure, i.e. manually. Most commonly, natural selection is applied where the individuals' fitness values are compared to select the best individuals for reproduction. Many selection procedures exist of which the most common and used in this dissertation are:

- Roulette wheel (Goldberg 1989): all individual fitness sums are stitched into an array like a stacked bar chart, giving a maximum value of the summed fitnesses. A random number between 0 – max(sum(fitness)) identifies an individual within the stack, as if a ball on a roulette wheel is spun and settles for a colour band. The larger the fitness of an individual the higher the chance of selection, increasing the probability to return the best individuals

- Rank: same as tournament but ordering the individuals' fitness values into a rank first and then drawing a random number between 0 and max rank
- Tournament: either selecting several random individuals and then singling out the best or selecting one randomly and comparing iteratively to other randomly drawn individuals until a winner is chosen
- Pareto fronts (Horn et al. 1994): when using multi-criteria evolution, no individual exists with all criteria optimized or dominating all other individuals. Hence, an archiving process over several generations is implemented that represent the number of non-dominated criteria. The first front (archive) includes individuals where none of the criteria are dominated by any other individual. The second front includes individuals with one dominated criterion, etc. Within each front there are several differently weighted individuals that all represent a trade-off between criteria. Each generation selects their best individuals and compares them to the fronts to replace (or not) an already archived individual (see 5.1)

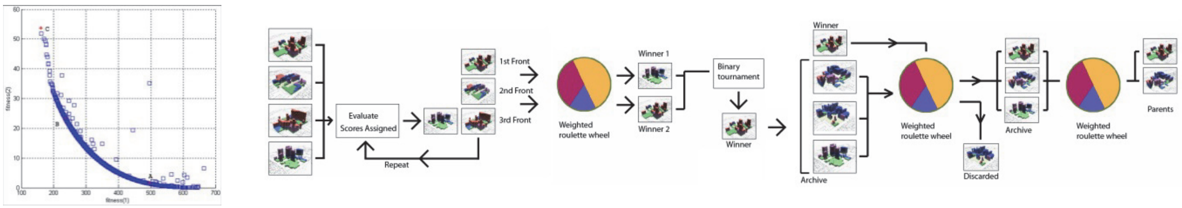


Fig41. *Pareto front (left) showing the many individuals of hundreds of generations that form a front; and (right) the complex selection and archiving process in a multi-criteria optimization (Finucane et al. 2006)*

3.1.4 Reproduction

Having selected individuals for reproduction, the actual evolutionary mechanism begins. There are two evolutionary operators: cross-over and mutation. In cross-over two individuals' chromosomes are split into *n* number of cross-over points and symmetrically swapped to create new chromosomes. Additionally, mutation is applied to the chromosomes by switching one of the binary bits of a chromosome to the opposite value (0=1/ 1=0). The mutation probability is not constant but needs testing and is relational to the size of the chromosomes and population.

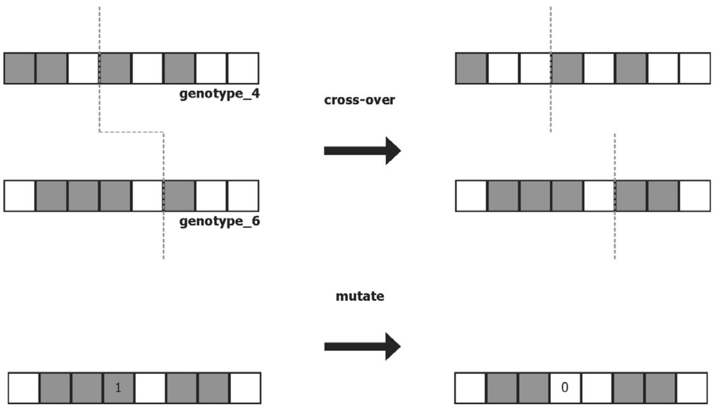


Fig42. *Evolutionary operators of the basic GA: (top) cross over at a single point splitting two chromosomes; (bottom) mutation of a single bit from binary state 1 to 0 (Derix 2008)*

The evolutionary mechanism and selection process are intertwined. Commonly, each individual in a generation is produced from a new set of parents. Steady-state optimization replaces a single individual in the population in each generation only. In small populations, only one father and one mother can be sufficient (Coates 2010). The selected parents are called genotypes and their offspring phenotypes.

```
PSEUDOCODE - GENETIC_ALGORITHM
{
   initialize
   {
      generation g = 0;
      encode embryology(g);
      generate initial random population phenotypes from embryology;
   }

   While (g < max_generations)
   {
      evaluate_fitness of population using fitness function;
      selection of fittest as genotypes;
      reproduce population as phenotypes from fittest genotypes ;
      {
         cross-over;
         mutate ;
      }
      g = g + 1;
   }
}
```

Fig43*. Generic genetic algorithm pseudocode used by author at UEL CECA between 2003-2008: generations g are used as halting function and two fittest individuals in a generation produce the phenotypes of next generation*

3.1.5 Tendencies

The combination of low permutation probabilities, small populations or limited 'parents', i.e. only one pair of parents for many individuals can lead to *genetic drift*, which causes quick convergence towards a limited variation within chromosomes (small gene pool), making all individuals similar (Nunes de Castro 2006). Another issue with GAs can be the convergence towards local maxima. All possible solutions within a design space are called the fitness landscape. If a population converges towards a peak (or trough) within a landscape it is difficult to extract itself from this apparently optimal location.

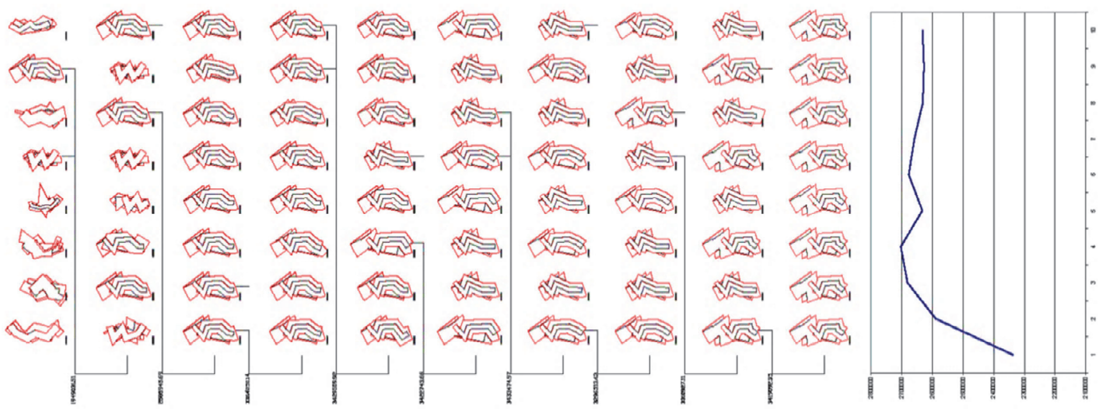


Fig44*. Evolutionary tree of a simple geometric assembly over ten generations (left to right – images rotated by -90°); because the population size is small (eight individuals) and only two selected individuals as mother and father breed a new generation, the population converges in only 10 generations towards some local maximum, representing 'genetic drift' (Coates 2010)*

3.2 RELATED SEARCH PROCEDURES

A GA is a population-based search meta-heuristic but other single individual combinatorial search meta-heuristics exists, which employ a similar generational stochastic approach. The most basic algorithm is *hill-climbing.* Hill-climbing can be done by various representations like agents, graphs or configurations. The agent-based analogy is most accessible: location *l* has a performance value like its z-axis value. An agent randomly choses an accessible location *l'* (like an adjacent cell on a mesh) and checks if *l'* has a higher z-axis value. If yes, then he moves to *l'*. If not, repeat random locations sampling for higher z-axis values. Due to its location based nature, hill-climbing is called 'local search'. In configurations, a position or element like a node on a mesh or array is randomly changed, similar to the mutation of a GA. If this change increases the overall performance of the configuration, it is accepted, otherwise rejected.

Another such stochastic search meta-heuristic is *simulated annealing* (SA). SA represents an analogy to metal or glass cooling processes and was first proposed by Nicholas Metropolis (Metropolis et al. 1953) and generalized for combinatorial optimization problems by Scott Kirkpatrick (Kirkpatrick et al. 1983). Similar to hill-climbing a discrete parameter in a configuration is changed and the new candidate configuration evaluated for performance. The key difference to hill-climbing is that if the candidate configuration is not better than the existing one, a probability function with a threshold determines whether the worse state is accepted anyway. This allows a larger exploration of the search space and enables the system to avoid local maxima. The threshold for accepting less well-performing candidates is lowered over time and the magnitude of change reduces, focussing adjustments to fine-tuning. This magnitude of change is called 'temperature' in reference to material annealing and the probability threshold represents the cooling. A configuration can still get trapped in local maxima locations of the search space when it happens to traverse them at a low temperature state.

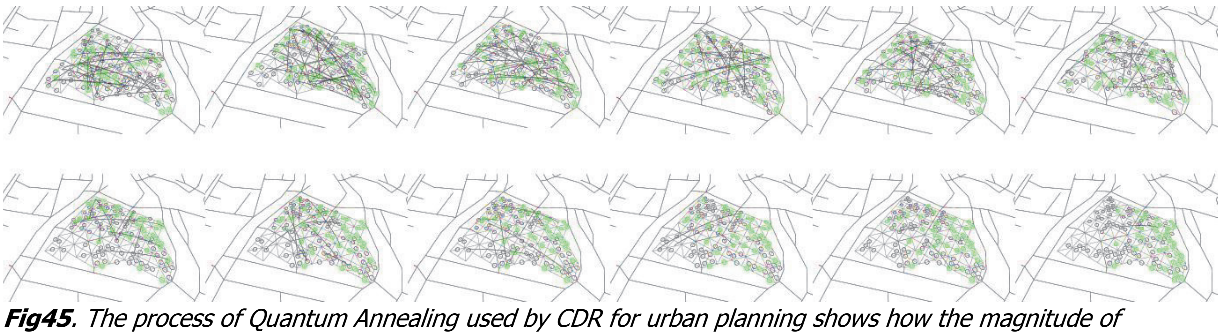


Fig45. *The process of Quantum Annealing used by CDR for urban planning shows how the magnitude of changes within a configuration reduces over time due to the reduction of the temperature variable which controls the probability threshold (Derix el al. 2012)*

There are many variations on the basic hill-climbing and annealing algorithms and a more complex annealing search meta-heuristic called Quantum Annealing is described in chapter 7.1.1, which like the GA is based on populations and manages to extract itself from local maxima (see 7.1.1).

3.3 AGENTS

Another design space search meta-heuristic is based on agent-based modelling. As the name suggests an *agent* provides an agency for somebody else, here mainly in a digital model as an agent of the observer. Stemming from computer science, agents exist as software programs of different types. The main two types used in architectural computing are either statistical agents or mobile agents. While statistical agents can be any software that evaluates data and takes some action, mobile agents are effectively 'moving' across some spatial abstraction, which Yehuda Kalay (2004) and the KbD community calls *Intelligent Agents*[30]. When populations of social agents are used, one speaks of multi-agent systems. In architecture, agents can be used for spatial analysis as well as generative design as demonstrated in the following chapters.

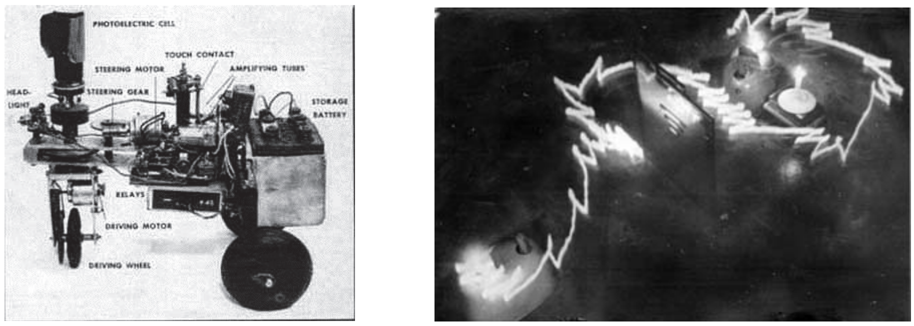


***Fig46**. Grey Walter (1950): Elmar and Elsie phototropic robot agents*

Mobile agents, which were predominantly used at CECA and CDR, started out as analogue robots invented by William Grey Walter (1950) at MIT in 1948. Walter built phototropic robot agents that evaluated light levels at each step (or motor axle rotation) by a light sensor, controlling the robots' heading and speed. Adding a light source to the robots would trace out the sub-space at which the sensors perceived the light levels set by a threshold, producing a contour map of the environment visible to the robots.

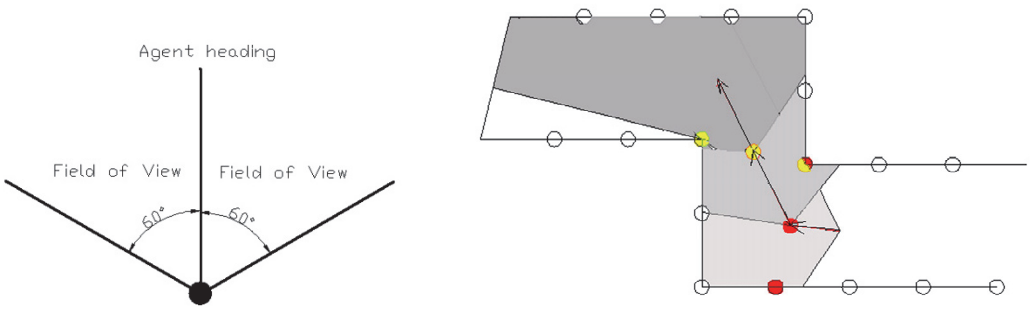


***Fig47**. Paul Coates' basic agent configuration with three direction vector probes (left); applied to a corridor by Junjie Shuo at CECA, 2005*

Paul Coates researched agent-based generative design at CECA from the mid-90s, using Craig Reynolds' (1987) *boid* algorithm. In order to simulate animal movement behaviours in spatial environments, Reynolds equipped his boids (bird-like objects) with virtual spatial probes for sensing. Additionally, boids have a direction and

---

[30] Agents that take their 'own' decisions subject to information from their context are variously called 'intelligent' or 'autonomous'.

speed. The cognitive structure of agents at CECA were generally into a configuration of a position vector with three direction vectors describing the field of view (FOV) in a plane at {– 60° = left , 0° = straight , +60° = right }, producing the typically assumed 120° human FOV (Coates and Schmid 1999). The FOV direction vectors produced a line at a set magnitude (usually the agent's *step-length*), which was used to probe intersections in the model, i.e. finding line – object (generally other lines) intersections, angles and distances to obstacles, which were used to adjust the heading. If more than one direction is available, either a random function or another constraint would determine the heading.

This basic configuration could be adjusted to sample various models of local perception of a user and informed design decisions as simulated builders.

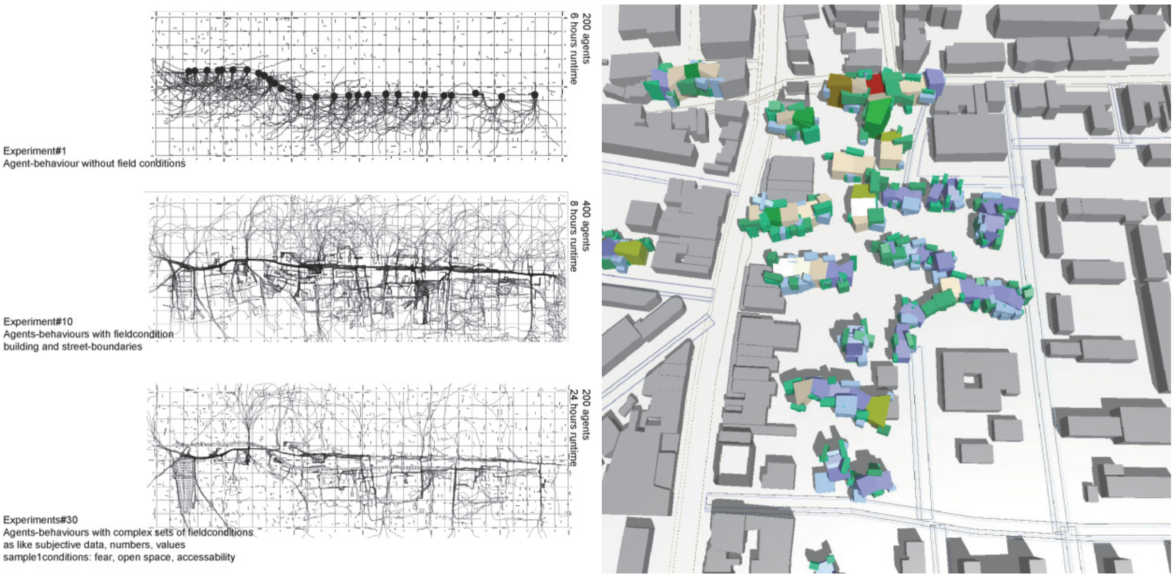


Fig48. *Two student projects at CECA (Coates and Schmid 1999): (left) Riko Sibbe's mappings of different spatial conditions by the same agent behaviours; and (right) Sean McMillan's urban densification; both projects set on Kingsland road in London*

Agents are also representative of the *embodied* computation paradigm because like Walter's agents, they represent geometric dimensions and interact with spatial environments. This was originally discussed by Seymour Papert (1980) who helped to developed the Logo programming language in 1967 to teach children an intuitive spatial way of computer programming. Papert invented *turtle graphics*, mapping the behaviour of turtle-like agents to model complex geometric phenomena (Papert 1980).

There are many examples of multi-agent systems that experiment with social behaviours of animals, which are used in design as meta-heuristics for optimization (Shea et al. 2006). For example, Reynolds developed the *swarm* behaviour algorithm (Reynolds 1987), while Theraulaz (Theraulaz and Bonabeau 1995) developed an environmental construction model based on social animal communication first described by Grassé as *stigmergy* (Theraulaz 1999); stigmergic systems also inspired

Marco Dorigo's ant-colony optimization algorithms (ACO) (Colorni, Dorigo and Maniezzo 1991).

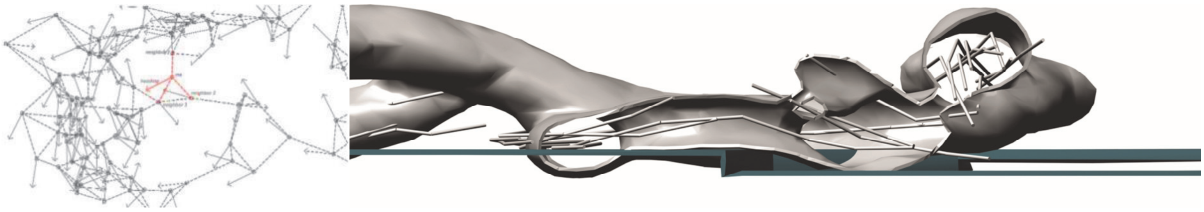


***Fig49**. Diagram of the swarming agents algorithm developed by Pablo Miranda in 3D as the first example of swarming in architectural design, 1999 (Miranda and Coates 2000); (right) a isospatial diagram of the swarm space by Miranda*

Many geometric phenomena can be described through non-intelligent agents such as the *random walks* by the 1950s German computer artist Frieder Nake who produced agent-based drawings using stochastic processes like Markov chains to generate graphical patterns (Kluetsch 2007). Markov chains are statistical sequences of random numbers that are constrained by a present number (state). For agents this can easily be translated into positions that determine future positions, similar to the probe-based heading of the basic CECA agent or Reynolds' *boid*. Also hill-climbing as described above can be represented through a Markov chain agent.

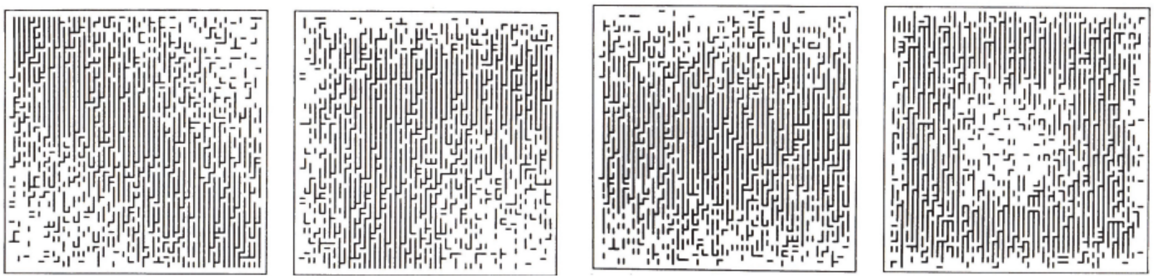


***Fig50**. 'Walk-through Raster' by Frieder Nake, 1966 (Kluetsch 2007)*

Paul Coates strongly believed in the representation of complex spatial phenomena via agent behaviours and experimented with a series of algorithms through agents like diffusion-limited aggregation (DLA), reaction-diffusion to generate Voronoi diagrams, attraction-repulsion forces (Coates 2004; Coates 2010) (also called force-directed graphs (Eades 1984).

```
PSEUDOCODE - BASIC_AGENT
{
    time t = 0;
    initialize agents a (direction, speed , FOV);

    While (t < max_steps)
    {
        For each agent {
        direction = check_obstructions (direction + FOV, direction, direction – FOV);
        speed = check_distances to obstructions;
        FOV = FOV +/- deviation from obstruction;
        position(t+1) = position(t) + speed * direction;
    }
    t = t +1;
}
```

***Fig51**. Generic agent algorithm pseudocode used by author at UEL CECA between 2003-2008: the direction was mostly randomized between FOV limits to guarantee variation in direction*

3.4 CELLULAR AUTOMATA

The self-replication machine called the Universal Constructor by John von Neumann at the beginning of the 1950s was based on Stanislav Ulam's discrete state machine called cellular automaton (CA) (Langton 1995). Ulam's abstraction into discrete cells made it possible to represent complex patterns purely as topological organizations and enabled the testing of transition rules between generations of pattern states. The discretization of space provides a distributed representation where local units generate global states without knowledge of them. The cellular automaton became popular through John Conway's Game of Life in 1970, which allows the simulation of pattern evolution based on a simple two state binary cell that represented 'death' = off or 'life' = on (Langton 1995). The discretized representation of the CA as spatial calculation grids became widespread in architectural representation as already shown above. Cedric Price's Generator Project with John Frazer and many of John Frazer's projects such as the Universal Machine were based on CA principles (Frazer 1995). Bill Hillier's generative settlement growth syntaxes are based on CA principles and Paul Coates' first algorithm of space syntax is a CA (Hillier and Hanson 1984). Michael Batty's Fractal Cities book became seminal for the simulation of the growth of urban patterns using CAs (Batty and Longley 1994). Paul Coates eventually also introduced the third spatial dimension to architectural design in 1996 (Coates et al. 1996), which provided the basis for some CA models in this dissertation (5.2-3; 8.1).

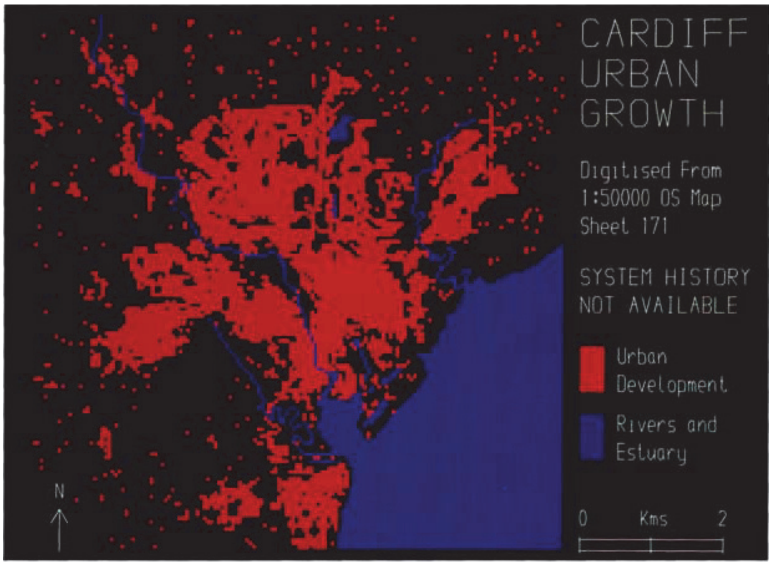


***Fig52**. Regenerating the urban growth pattern of Cardiff, UK, simulated with a CA (Batty and Longley 1994)*

3.4.1 Ontology and Algorithm

A CA consists of discrete finite state cells, i.e. numeric units whose values turned into discrete states. Cells are commonly arrayed into one dimensional series or two dimensional grids with an orthogonal cell shape. With a traditional orthogonal grid, the number of adjacent cells that form the neighbourhood is usually the original van Neumann neighbourhood of four cells or the extended Moore neighbourhood of eight cells, which takes the diagonal cells also into account. Coates' spatial grid has a von Neumann neighbourhood of six and a Moore neighbourhood of 26 cells. One can however implement any kind of topological neighbourhood.

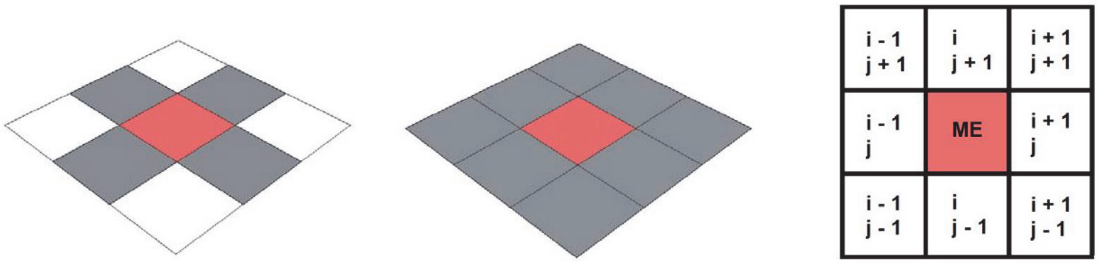


Fig53. *CA cell neighbourhoods: von Neumann (left), Moore (middle) and the 2d array indeces of a Moore neighbour (right)*

Traditionally, the cell state is represented by a single colour but the author introduced geometric representations of states at CECA that led to any number of architectural CAs (see 5.3). The cell state is calculated by a function called the *state transition rule*. The transition rule takes into account the states of the cells within the neighbourhood of a cell and produces a state value which is compared against the state of the calculating cell. The simplest rule is the *voting* or majority rule, which simply averages the states of all neighbouring cells like a consensus. The calculating cell then compares its own state and depending on a set threshold adopts the state of the majority or not. A transition actually only takes place if the cell has to change state, i.e. if the conditions are fulfilled for an adjustment. As the state transition rule applies to all cells, it is a global function, which can contain as many conditions as cell states to catch all possible variations between calculating cell and neighbourhood.

Not all CAs work with simple integer states like the voting rule or Conway's Game of Life, which only had two integer states (on = 1/ off = 0). More complex state values exist that are based on real numbers, weighted percentages or whole matrices, allowing for more heterogeneous representations (see 5.1+3).

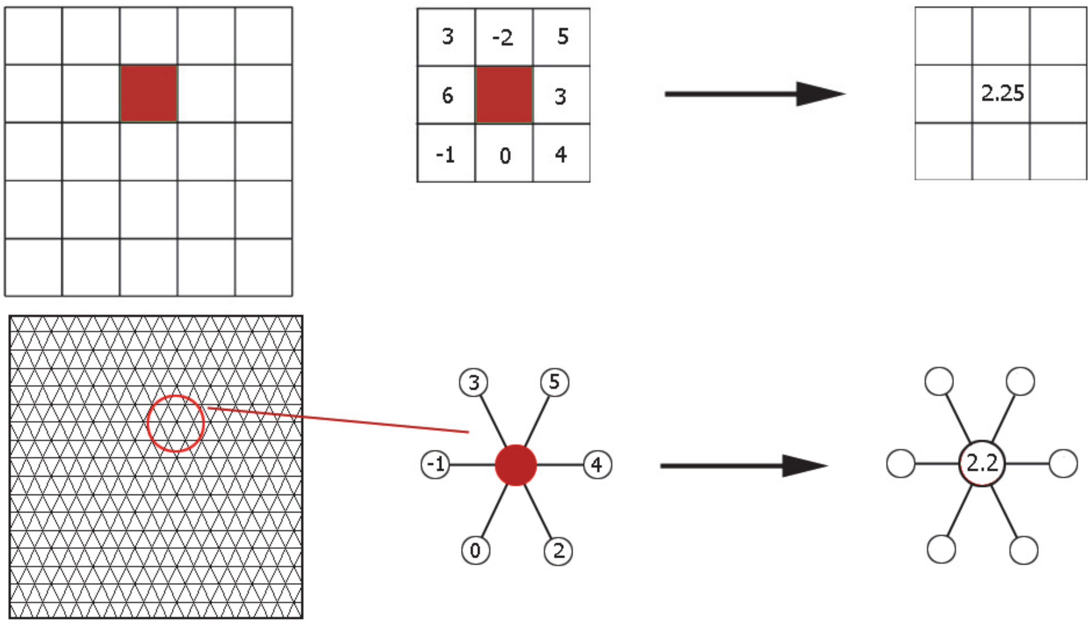


Fig54. *CA topologies: (top) the standard orthogonal 2d grid topology and how to calculate the red cell's average state value from its neighbours' values; and (below) the same done for a triangulated grid with hexagonal neighbourhood, resulting in a different cell state value*

It is critical that cells calculate their state transitions synchronously and not in sequence. To do that a generation has to be frozen where each cell calculates its future state and stores that state before updating. Only when all cells have executed the state transition rule and calculated a future state, must all cells update their states from present to future concurrently.

```
PSEUDOCODE - CELLULAR AUTOMATON
{
    initialize
    {
        time t = 0;
        set transition rules ;
        set cell states randomly;
    }

    While (t < max_time)
    {
        For each cell
        {
            calculate neighbours values at t;
            evaluate cell state based on transition rules against neighbourhood_value;
            update cell states:
            If (rule r1 and/or rule r2 and/or rule r3 etc apply OR cell_state > threshold)
        {
            future_state = rule_state OR neighbourhood_value;

        } else {
            future_state = current_state;
        }
        For each cell
        {
            current_state = future_state;
        }
    t = t +1;
}
```

***Fig55**. Generic cellular automaton algorithm pseudocode used by author at UEL CECA between 2003-2008 the state transition can be governed by complex rules or by value thresholds like in the voting rule; the synchronous state comparison is done using a 'future_state'*

3.4.2 Variations

The representational qualities of the CA as synchronous, discrete and distributed field enables the simulation of many complex patterns through a bottom-up rule-based process, ideal for algorithmic exploration. In architecture, CAs can be used for relaxation through diffusion (which is again a voting rule), partitioning through reaction-diffusion and the flood-fill algorithm (see 7.1.1), or branching through diffusion-limited aggregation, to name but a few. In urban planning, CAs have had a long history such as Batty's urban land-use models (Batty and Longley 1994) or Coates' space syntax model.

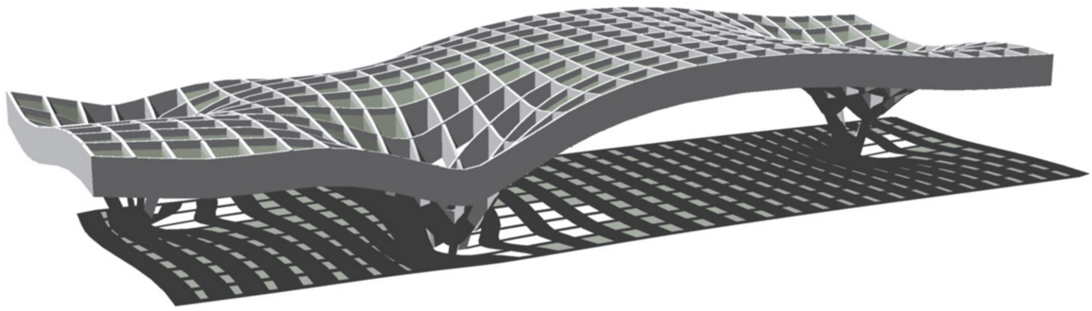


Fig56. *An example of relaxation by diffusion using a CA by the author as teaching material at CECA, 2007*

3.5 GRAPHS AND NETWORKS

Many computational representations are calculated using graphs, employing graph theoretical algorithms or simply as topological visualization. In graph theory, a graph represents a *connection* between two discrete elements, which are usually called *nodes* or *vertices*. The connection is called an *edge*. The edge represents some exchange of information or relationship between the pair of nodes and can be bi-directed or directed. Most graphs used in this dissertation are bi-directional because they serve as a representation of circulation, routes or paths. Graphs generated from building adjacency matrices are directed as connections from one room to another.

When graph edges represent non-topological variables like geometric or geographic dimensions or any other magnitude, they are called *weighted*. Un-weighted graphs represent topological pairwise connections. A graph connecting several nodes can either be analysed by calculating on the weighted edges or number of topological connections (Nunes de Castro 2006). There are two basic distinctions of graph connection structures: a *tree* or a *network*. A tree graph contains connected nodes without *cycles*. Cycles are loops of connected nodes that close into circular paths. In tree graphs there is always only one path between a connected pair of nodes. Network graphs contain cycles and thus there are at least two paths between a pair of connected nodes. A subgraph represents a tree graph that connects all nodes without being most efficient, called *spanning tree*. Efficiency is calculated by some cost function of weighted edges, such as the *minimum spanning tree (MST)*, which represents the shortest connection between a set of nodes (Prim 1957).

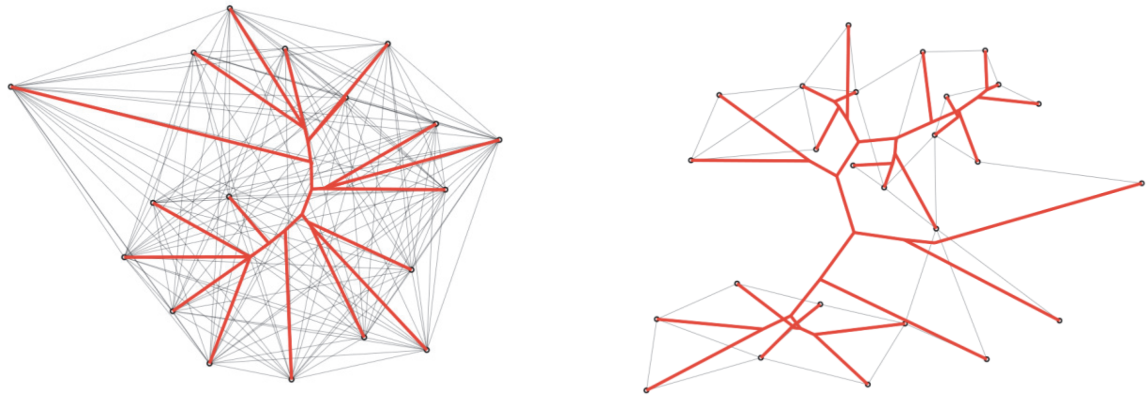


Fig57. *Two spanning trees as sub-graphs developed during the Fucon project (6.2)*

Some standard measures exist to evaluate the topology of a graph or network, first invented for sociological network analysis (Freeman 1977; Wasserman and Faust 1994). Bill Hillier introduced some of those measures to architectural theory through the *Social Logic of Space* (Hillier and Hanson 1984). The core measures are based on the principle of *centrality*, indicating the importance of nodes within a network:

- *Degree*: number of edges of a node (connectedness)
- *Closeness*: distance of a node to all others, indicating the geometric centrality of a node (proximity)
- *Betweenness*: strategic centrality within a graph, indicating the number of shortest paths a node is allocated on (through-movement)
- *Cycles:* number of cycles or looped paths a node is allocated on (choice)

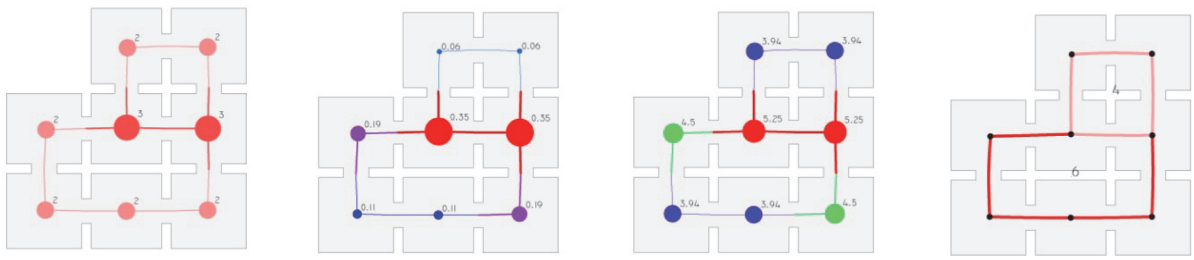


Fig58. *Four basic network centrality measures from left: degree, closeness, betweenness and cycles (Deleuran and Derix, 2013)*

While previously, network measures were mainly attributed to nodes (locations), edge centrality measures are equally important in architecture as they indicate flows along a path. Paolo Crucitti and colleagues proposed the usage of edge centrality measures instead of space syntax's axial maps, also suggesting other measures like *straightness* (Crucitti et al. 2006), later called *angular* (Turner 2000). Hillier proposed the graph measure of spatial *depth*, akin to node closeness, to illustrate the correlation between social and spatial organization. His *justified graphs* are built on closeness centrality, adjusting a graph around a starting node and ordering the rest of graph's nodes by their *depth* (Hillier and Hanson 1984).

3.5.1 Algorithms and Types

The *betweenness centrality* network measure illustrated above requires a traversal algorithm to select between many different possible paths through the graph to determine the shortest for each pair of nodes. There are two primary search algorithms: depth-first (DFS) and breadth-first (BFS) search (Kodicek 2005). The DFS traverses a graph from a starting node along connected branches, backtracking when coming to the 'end' of a branch. Visited nodes are marked to avoid repetition along previously visited branches. The procedure is similar to recursive branching. BFS on the other hand, traverses a graph from a starting node along its connected nodes first (right-left or left-right), using topological levels instead of branches. When all neighbours have been visited, the next level of connections is visited, traversing sideways instead of downwards as in the DFS. To keep track of nodes' connectivity, a list is generated during execution called a queue.

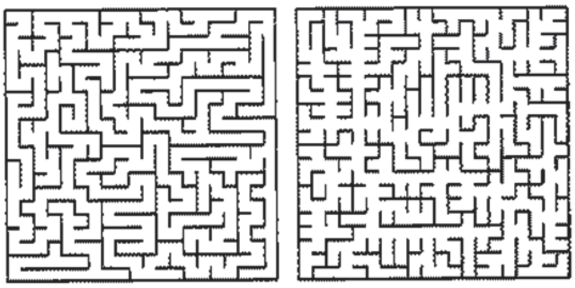


Fig59. *Maze generation algorithms (Kodicek 2005): using DFS and backtracking (left) and Prim's algorithm using a stochastic edge selection*

Both search algorithms are repeated for all graph nodes, each compiling a list for all paths to all nodes, which can now be weighted for some cost function. The cost function evaluates for an efficiency criterion such as least number of nodes for topological distances. To find the shortest path between pairwise nodes, the non-negative edge length has to be taken into account, summing up all edge length for all paths. The smallest sum is the shortest path. While both algorithms can be used to search for paths, each has qualities that lend themselves for different situations such as described in favour of DFS for *maze generation* in 5.3 for the VITA shelving or in favour of BFS in 6.1 for the Visible Polygon Traversal Algorithm (VPTA).

The best known algorithm for calculating the shortest distance between two nodes is the Dijkstra algorithm, after Edsger Dijkstra (1959). Dijkstra's algorithm requires edge weighting such as length or slope for the cost function, and represents a hybrid between BFS and DFS, because it first evaluates the cost of topologically connected neighbours but secondly traverses branches as sub-graphs. All edge weights from a starting node (parent) to its neighbours are evaluated and the neighbour with the lowest cost (say shortest distance) is chosen to be the next parent node. From there, all edges are evaluated for their weight and added to the previously smallest weight, giving a new sum along each alternative edge. From these projected sums, the most efficient is chosen to traverse. Repeating this procedure, a list of nodes is assembled with the smallest sum of edge weights towards a target node, selecting the shortest path between a pair of nodes.

```
PSEUDOCODE – DIJKSTRA_SHORTEST_PATH
{
    initialize
    {
        select origin node;
        queue set all nodes as unvisited;
        For each node (set set distance to origin to some maximum value);
    }

    While (queue not empty | destination node not reached)
    {
        search neighbourhood of parent node by calculating edge distances to all topological neighbours;
        sum previous path length with all alternative neighbourhood edge lengths;
        select alternative path with shortest path combination (cost);

        replace previous shortest path with new combination;
        queue set new node of shortest combination as parent, remove from queue and add new topological neighbours;
    }
}
```

***Fig60**. Generic Dijkstra shortest path algorithm pseudocode used by CDR. The queue is a priority queue that sorts topological neighbours by distance*

3.5.2 Visibility Graph

In order to calculate paths as sub-graphs, a graph has to be available. In spatial applications, paths relate to movement between obstacles. In robotics, graphs are generated from intervisible nodes of obstacle polygons (in plan), called *visibility graphs*, to plan the motion paths (O'Rourke 1994). Visibility graphs are simply constructed by inserting bidirectional edges between nodes – both between polygons and of the same external perimeter of a polygon - that can 'see' each other, meaning that the edge does not intersect a polygon perimeter.

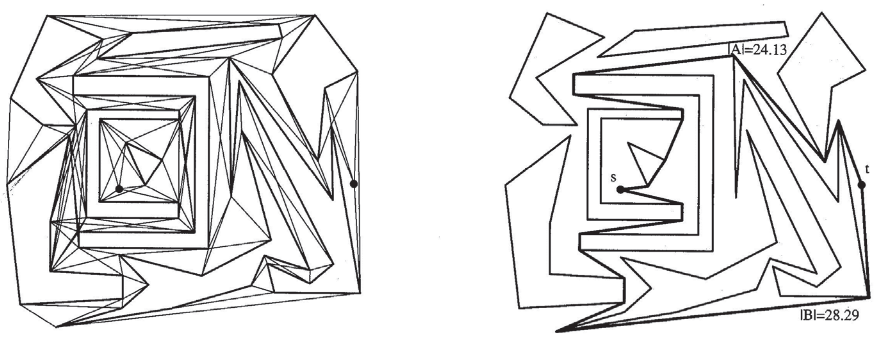


***Fig61**. Visibility graph construction (left) by Joseph O'Rourke (1994)for motion planning for robot paths; and the shortest path between point s and t using Dijkstra (right)*

Using the visibility graph, a host of efficient paths can be calculated using Dijkstra's algorithm with different weights, such as edge length for shortest metric path, degrees between edges for least angular distance (Turner 2000) (see 6.1) or least turns for topological distance.

In 2001 Alasdair Turner developed an architectural version of the visibility graph for spatial analysis, which he called Visibility Graph Analysis (VGA) (Turner et al. 2001). Instead of evaluating the intervisiblity of nodes of obstructing polygons and connecting them into a graph via edges, discrete grid nodes are evaluated and connected by edges. So, instead of obstructions as a construction scaffold, discretized permeable areas provide the construction scaffold, allowing for the quantification of usable space. The visibility graph becomes densely connected and provides an analysis field for multiple measures that Turner proposed such as

- *Neighbourhood size*: the number of connected nodes representing the area of the view-shed
- *Clustering coefficient*: measure of convexity of boundary, indicating how many separate areas exist within a view-shed that cannot see each other
- *Mean shortest path*: number of edges to traverse to get from a departure to an origin point (least number of nodes); this is equivalent to topological turns to reach spaces and therefore similar to spatial depth

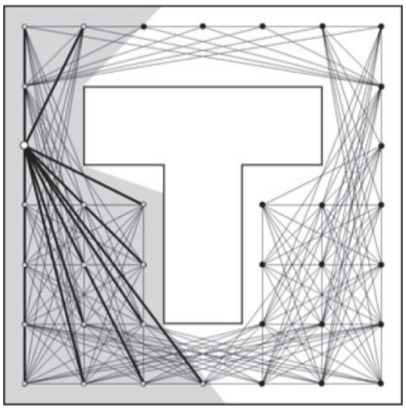


Fig62*. The visibility graph constructed from a discretized area grid by Alasdair Turner (Turner et al. 2001)*

3.5.3 Isovist

The VGA was developed to allow the measuring of spatial fields between viewshed polygon edges. The analysis of the 'field of view' or viewshed was introduced by Michael Benedikt (1979) at the University of Texas at Austin with his concept of the *isovist*. Benedikt's isovist was constructed from radial lines emanating 360 degrees from a viewing location in two dimensions and does not constitute a graph. The endpoints of those radials would connect into a perimeter shape, called the Isovist as all the vertices on the perimeter can see the view point (*vantage point*).

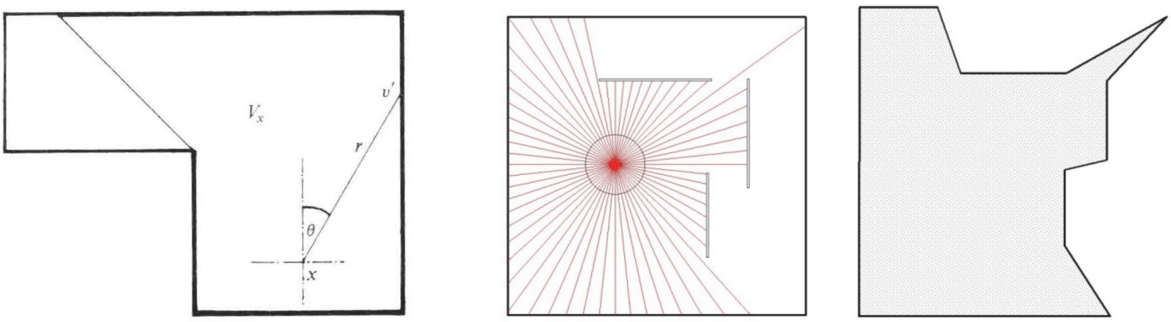


Fig63*. Michael Benedikt (1979) Isovist definition and two diagrams by CDR for the National September 11th Memorial Museum (Derix et al. 2008) showing a radial lines construction through projection and its resulting isovist (middle and left)*

The isovist consists of three measurable entities: radials, perimeter and area. Benedikt's measures include:

- *Area*: visible area from isovist (not as number of vertices as in VGA)
- *Real-surface perimeter*: perimeter length of view, meaning the elevation edges
- *Occlusivity*: ratio of seen versus unseen portion of perimeter (6.1)
- *Compactness*: through *variance*, *skewness* and *circularity* of the radials. These three measures were evaluating the distribution of radial lines and perimeter line, not describing a relation to the vantage point, and have seen little application so far
- *Minkowski model*: sequence of isovists constructing a two dimensional measure of space/time
- *Isovist Fields*: Benedikt anticipated Turner's spatial evaluation by proposing that a space should contain an array of isovists that generate contour maps from their above measures, showing *isolevels* of spatial properties

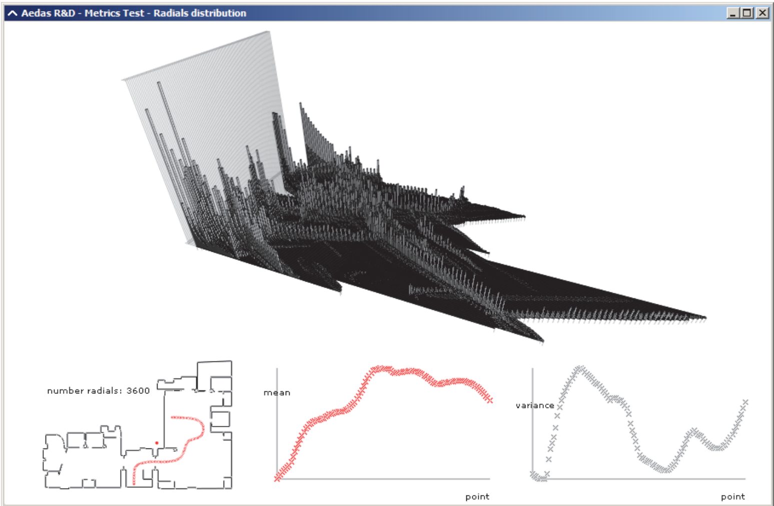


***Fig64**. Movement isovist, called the Minkowski model, showing viewsheds along a path and their properties; developed by Aedas CDR for the RIBS projects, 2010*

The shape constructed from radial lines however ignores the space between the perimeter and the view-point and thus required mathematical descriptions for its evaluation, separating the analysis from the intuitive generative representation, which Turner's VGA remedied.

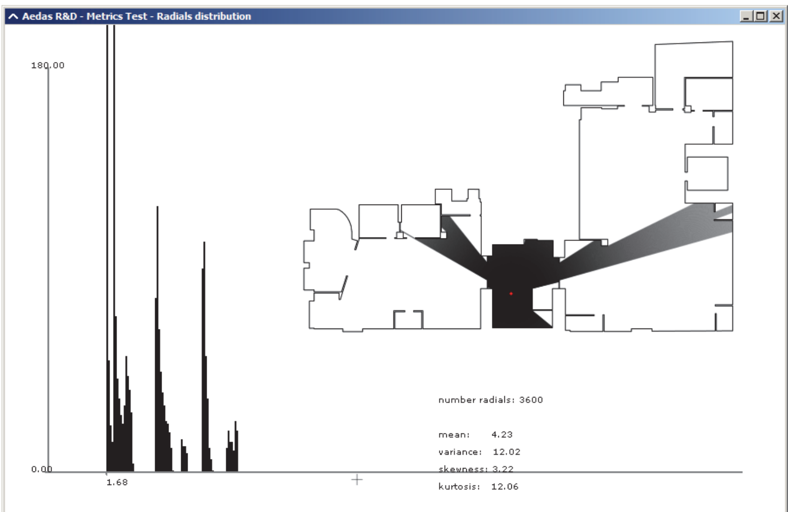


***Fig65**. Single isovist from the red node, showing four of Benedikt's measures and a distribution histogram at the bottom of radial lengths; developed by Aedas CDR for the RIBS projects, 2010 (6.1)*

3.5.4 Topological Skeleton

The topological skeleton of a polygon – with or without holes – represents a dimensionality reduction of a shape. For a two-dimensional polygon a one-dimensional graph is extracted, for a three-dimensional volume a two-dimensional surface (Leymarie and Kimia 2008). The reduced dimensional representation constitutes the central topological skeleton of the higher-dimensional shape, both geometrically as the central distance between polygonal edges as well as topologically because the skeleton is produced from adjacent edges. For architecture and urban planning, the reduction of polygonal shapes into tree or network graphs provides an invaluable source of automatic spatial structure generation for network analysis (Batty and Rana 2004; Franz and Wiener 2008), used in this dissertation.

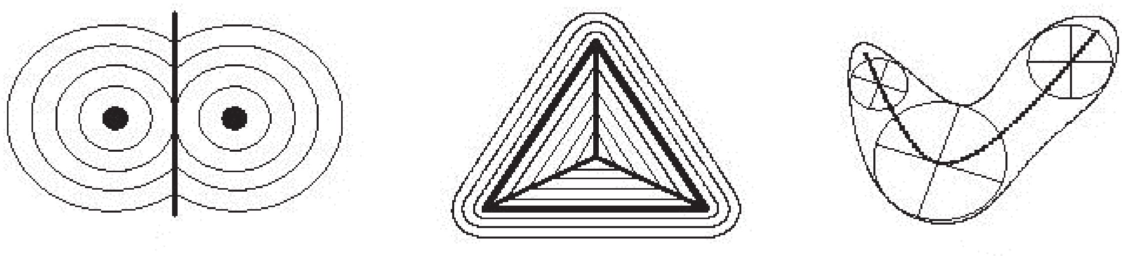


Fig66. *Shape skeletons or Medial Axes image from (van Tonder 2004)): three modes of producing medial axes as proposed originally by Harry Blum (1967); (left) blocking waves or wavefront algorithm; (middle) same algorithm on polygon producing the medial axes skeleton along the ridges of the 'offsets'; and (right) the tangent largest circles approach*

There are three basic construction algorithms for topological skeletal, namely the *straight skeleton*, the *Voronoi* method and the *medial axis*. In principle, topological skeletons represent distance fields where the edges of the resultant graph are equal distance between two or more contextual edges (Blum 1967). Nodes joining edges of the emerging graph represent a change and possibly number of constructing edges. The mathematical construction of the skeleton is called *medial axis* and is generated from circles that touch two or more edges (tangent circles). The collection of all circle centres produces the skeleton.

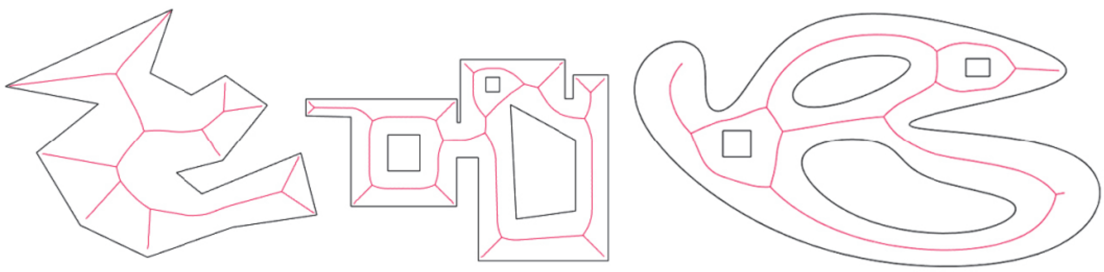


Fig67. *Three examples of a topological skeleton using the Voronoi method (Deleuran and Derix 2013)*

The straight skeleton differentiates itself from the medial axis by parallel offsetting straight lines from polygonal edges. The offset lines are bound by the angular intersections formed from two adjacent edges. At a given offset distances all adjacent edges intersect and produce the vertices for a shrunken polygon. The intersection vertices represent the ridges of the emerging straight skeleton. Where more than two edges intersect, a skeletal node is produced and a change in direction of the emergent graph edge occurs. Aichholzer and Aurenhammer (1996) describe the exceptions and hierarchical alignment of the skeleton for when non-adjacent edges intersect and produce two isolated offset polygons.

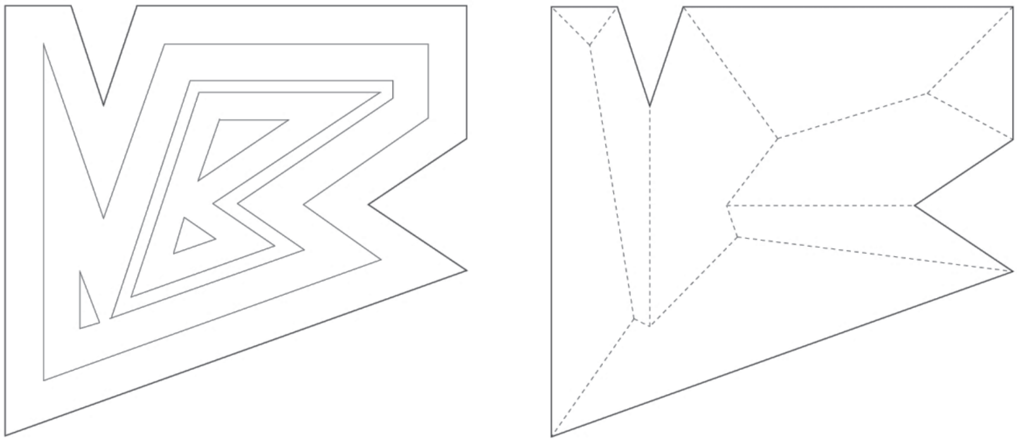


***Fig68**. Nested offset polygons through the shrinking process and the resulting straight skeleton (Aichholzer et al. 1995)*

While the medial axis and straight skeleton constitute continuous processes of construction, the Voronoi method uses discrete points along a polygon's perimeter for construction (Fortune 1987). As a distance field, a Voronoi partition represents the centre line between two vertices, rather than edges, similar to the *wavefront* algorithm (Blum 1967). More than two vertices produce centre line skeleta where more than three equidistant vertices produce a converging node or branching location with valence >=3. Emerging Nodes are then ordered by their distances and nodes of shortest distances are connected by an edge. Subdividing polygonal perimeters into equally spaced vertices provides the basis for the Voronoi method for generating a topological skeleton, which means that the Voronoi method is dependent on resolution: the higher the resolution, the closer it approximates the medial axis (Fabbri et al. 2002). Voronoi centre lines that lie outside the generating polygon are clipped. The graph is generated that connects the resulting Voronoi vertices of valence 1 or 2 (vertex with one or two connections) to graph nodes of valence >= 3, using polylines as graph edges (Deleuran and Derix 2013).

The advantages for spatial representation and computation are discussed in 6.2 *Generic Network Behaviour*.

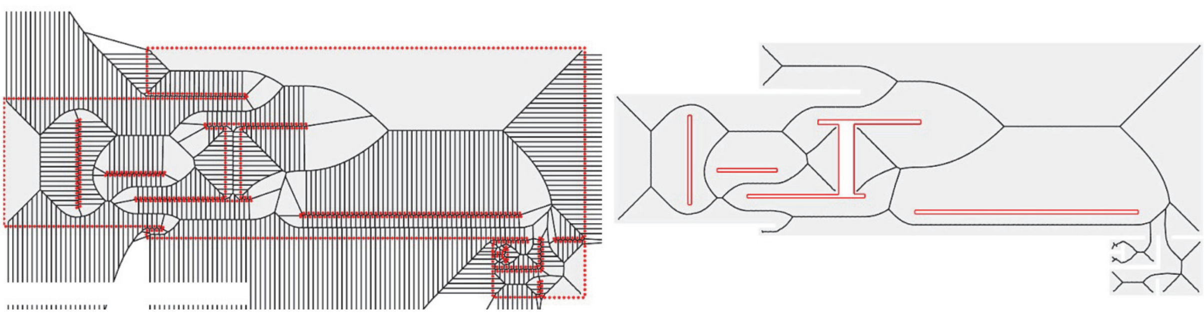


***Fig69**. A Voronoi diagram generated from a subdivided floorplan (Deleuran and Derix 2013) using Franz and Wiener's (2008) test layout: (left) subdivided polylines and their partitions and (right) the topological skeleton of the building plan*

3.6 SELF-ORGANIZING FEATURE MAPS

Another core algorithm for distributed representation used in this dissertation (6.3 7.3) is the self-organizing feature map (known as SOM), developed by Teuvo Kohonen (1981). The SOM belongs to the class of artificial intelligence techniques used for pattern recognition that are called artificial neural networks (ANN). Before Stephen Grossberg (1976) introduced the first mathematical concept of adaptive pattern classification, all ANNs constituted supervised models where the pattern to be learned is given as a goal. Instead, the SOM is an unsupervised learning network aiming to find goal patterns through self-organization. Traditional supervised ANNs consist of neurons (or nodes) that calculate a sum of numeric input values called *weights* and pass on a value if a threshold has been reached that is generated by a function called the *activation function*. Derived from the biological analogy of the human cerebral cortex, the input weights represent *synapses* and the output represents an *axon* connecting to another neuron. Due to the connected nature of distributed processing nodes, ANNs provide the mathematical model for the conceptual theory of *connectionism* (see 2.4).

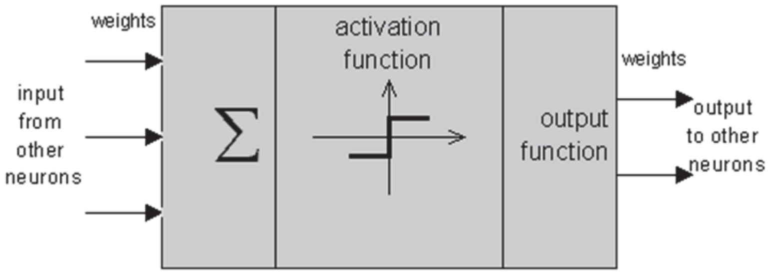


***Fig70**. Perceptron or neuron diagram, showing summed input (left), the activation function to calculate whether the sum crossed a set threshold for firing (middle) and the output values being passed on (Froehlich 1996)*

The first ANN was invented by Frank Rosenblatt (1957) and was called a Perceptron. The Perceptron constitutes the most basic threshold node and has continued to be the neuron architecture of choice, even for Kohonen's SOM. The Perceptron represents the essential threshold function (*activation function)* by creating a product from input weights, which has to be higher than a value to 'fire' and pass on a value. Marvin Minsky and Seymour Papert extended the basic perceptron, which consisted of a single layer only, into multi-layer perceptrons with more complex feedback mechanisms allowing values to propagate not only forwards (feed-forward networks) but also backwards (back-propagation networks). The forward or backward propagation changes the weighting along the synapses of the connections between neurons called the *weights*, which serve to *bias* the calculation of the *weighted sum* at each neuron, thus facilitating or inhibiting information flow. The adaptation of all synaptic weights for all network connections in order to calculate desired goal states from any input sample is called 'learning'.

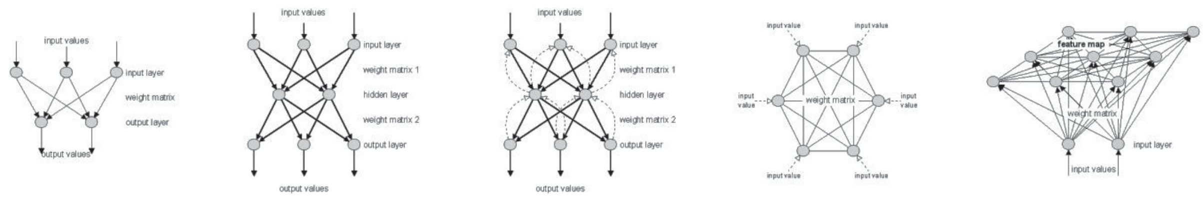


***Fig71**. Five ANN network types in chronological order: single layer perceptron, multi-layer feed-forward perceptron, back-propagation network, Hopfield self-organizing network and the Kohonen SOM (Froehlich 1996)*

Self-organizing ANNs apply a competitive or reinforcing learning method, which was discovered by the psychologist Donald Hebb and hence mostly called Hebbian learning (Hebb 1949). Contrary to supervised ANNs where neurons of a layer only connect to neurons of other layers, the two main self-organizing ANNs - the Hopfield and Kohonen networks (Froehlich 1996) - represent single layer networks where all neurons are interconnected between themselves and to all input samples. All neurons compete for each input sample and only one neuron 'wins' the competition. The winning neurons become specialized for specific pattern and reinforce their connection, increasing the strength of the synaptic weights, thus preventing others from connecting to specific inputs. Competitive learning hence generated network areas specialized for inherent data pattern. Those areas are called *perceptive fields* (Kohonen 1995) and are equivalent to pattern classes.

Perceptive fields as in the SOM are a reduction of dimensionality from an n-dimensional input space into generally a two dimensional map representation where topological patterns are easily accessible. The author introduced self-organizing ANNs to the field of architectural research in 1999 (Derix and Thum 2000) and also produced the first three dimensional representation (6.3).

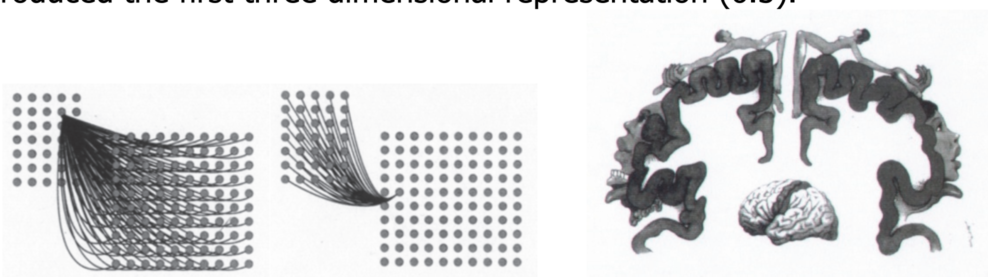


***Fig72**. The SOM has an input layer and the network map. All input samples are connected to all map neurons and vice versa, making the map neurons compete for input samples (Kohonen 1995); the Homunculus image shows which sensory input surface of the human body is mapped by intensity across the cortex*

3.6.1 Ontology and Algorithm

Officially, the SOM is described to have two layers with the first being the input layer and the second the network layer or map where the learning takes place. The topology of the classic SOM is fixed, meaning that all connections within the map are predetermined by their layout, commonly an orthogonal lattice with a von Neumann or Moore neighbourhood. Input samples are usually formatted as vectors containing Booleans, binary or real numbers. Neurons in the map need to reflect the input format to be comparable.

The map learning stage is called *training*, which initiates the competition between map neurons for input samples, calculating their vector difference, called *vector quantization* (Kohonen 1995). The standard learning function uses the *Euclidean distance* between vectors, calculating the difference between the input and the map vector scalars separately to produce a measure of similarity. But for finer differentiation the *dot product* can be used, normalizing all vectors before calculating the difference in direction between input and map neuron vectors. For binary vectors the Hamming distance is used, which compares each position of two vectors and outputs the number of unequal binaries as an integer value.

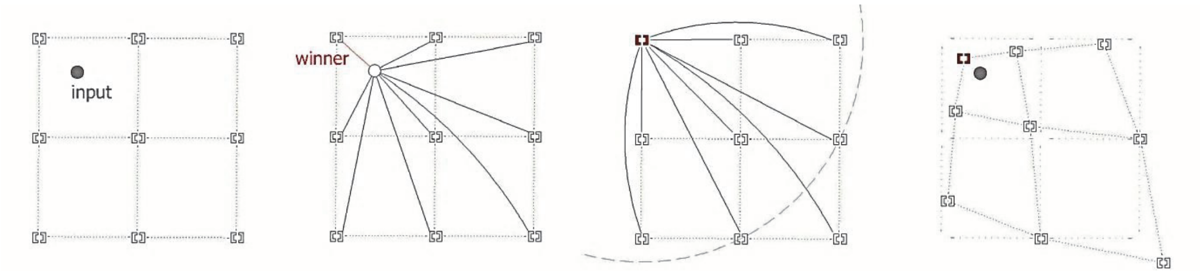


***Fig73**. A geometric illustration of map neurons competing for an input vector sample by calculating the Euclidean distance to find a winner who adapts his neighbourhood to the input sample by reducing the vector difference (Derix 2004)*

The map vector that is least different to an input vector is called a *winner* neuron. There are as many *winners* as there are input samples. When all input samples have found a winner, they organize the map by adapting first their own neural weights, i.e. the vector scalars, and then the weights of the neurons within their neighbourhood. Neighbourhoods are determined by the topological distance to a winner and the feedback from winner to neighbourhood neurons is inversely proportional to their topological distance. The adaption of weights is decreasing the Euclidean distance (difference) between input sample vector scalars and map neurons. The amount of difference reduction is determined by a learning coefficient, which again is proportional to the inverse distance to the winner. Hence, the winner adapts his weights (scalars) to the input sample more than the neighbourhood neurons that with increasing distance are adapted less (excitation) or beyond a certain threshold receive even negative feedback (inhibition). Since map neurons can form part of several winner neighbourhoods, a temporary sum of their adapted weights is calculated averaging all feedback before updating their weights.

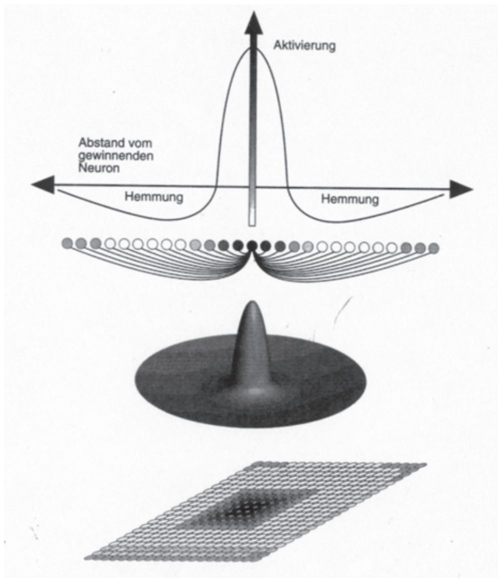


***Fig74**. An illustration of the topological neighbourhood and the distance-proportional feedback from winner to neighbourhood; this distribution is called the 'Mexican head function' (Froehlich 1996)*

When one generation of learning has been completed and the map neuron weights (vector scalars) updated, the learning coefficients and the topological neighbourhood radius variable are monotonically reduced before starting the competition again. When the values of the learning coefficient and the neighbourhood radius variable approach zero, the map stops training and the final classes emerge. With each generation, the *winners* establish stronger connections through smaller Euclidean vector distances to their input samples, generating and dominating *perceptive fields* within their map neighbourhoods.

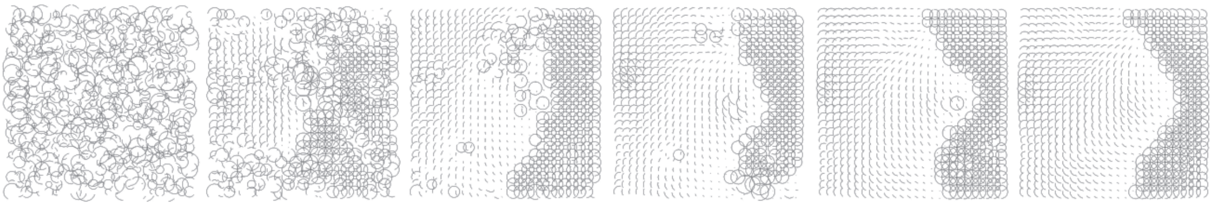


Fig75. *Five training generations of circles organizing themselves in a 2d SOM by opening segments and radii; teaching script by author at CECA, 2006*

Kohonen adaptation function using Hebb's learning rule is also commonly summarized as

$$Wij\ (t+1) = Wij\ (t) + Kd(t)\ [X - Wij\ (t)]$$

Where W is a node at grid position *i* and *j*, K is the learning coefficient dependent on the topological distance *d* to the winner, X is the input sample (Derix 2006).

```
PSEUDOCODE - SELF-ORGANIZING_MAP
{
    Initialize
    {
         time t = 0;
        set network node vectors (or weights) randomly;
        generate input samples i;
        set learning values (winner learning coefficient wl, neighbourhood learning coefficient nl, neighbourhood radius r);
    }

    Organize map:
    while (r < min_radius)
    {
        For each input sample i;
        {
            For each network node
            {
                evaluate Euclidean distance  = vector difference;
            }
        select winner_node(i)
        }

        Learning:
        For each winner_node(i)
        {
            evaluate topological distance to neighbourhood nodes;
            If (distance < r)
            {
                excite node by reducing Euclidean distance of weights to input_sample by nl * distance;
            } else {
                inhibit node by increasing Euclidean distance of weights to input_sample by nl * distance;
            }
            adapt winner_node(i) by decreasing Euclidean distance of weights to input_sample by wl;
        }
    }

    update learning coefficients (nl, wl, r) by t;
    t = t +1;
}
```

Fig76. *Generic Kohonen self-organizing feature map algorithm pseudocode used by author at UEL CECA between 2003-2008*

3.6.2 Variations

Kohonen realized that a fixed topology might be limiting because map neurons are distributed evenly across the input sample space. Details of the input space might be lost and hence he proposed an adaptive topology SOM such as the *dynamically defined neighbourhood* or the *growing SOM* (Kohonen 1995, p164 ff). Various researchers have subsequently developed this idea into more flexible and robust network models, most prominently Thomas Martinetz and Klaus Schulten's *neural gas* (Martinetz and Schulten 1991) and Bernd Fritzke's *growing neural gas* (Fritzke 1995) (see 6.3).

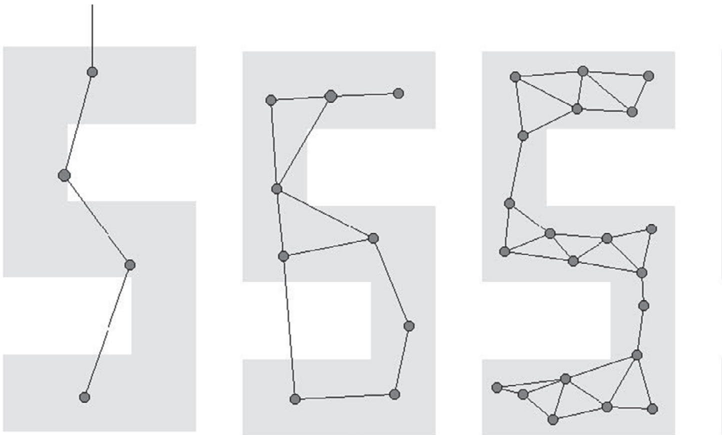


***Fig77**. Growing Neural Gas experiment by Philip Langley at CECA, 2007, using Fritzke (1995)*

Kohonen did not pursue the development of growing networks because he realized that the SOM represents an ideal generalizer while adaptive topologies might represent local details better but can oversee global patterns (Kohonen 1995). Because the SOM is a topological network, the resulting classification maps vary between training sessions.

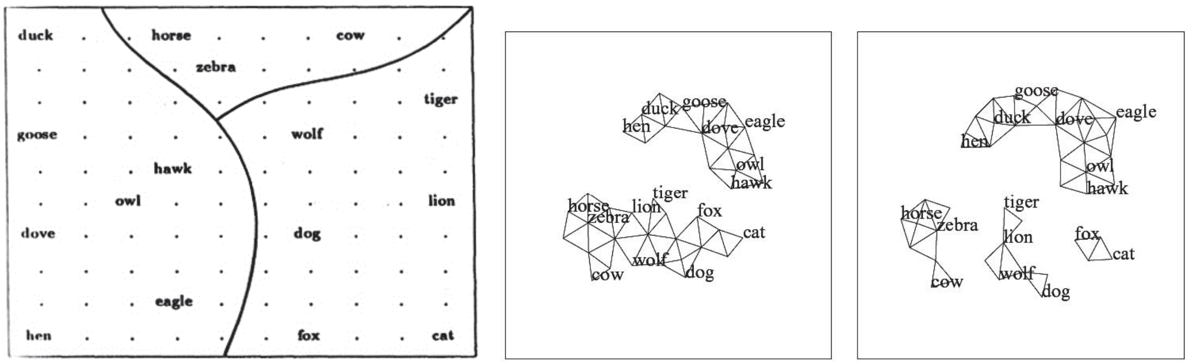


***Fig78**. Growing Neural Gas experiment by John Harding at CECA, 2008, using Fritzke (1995) on Kohonen's (1995) original animal features map (left)*

4 FROM OBJECT CONFIGURATIONS TO FIELD ORGANIZATIONS[31]

From Object Configuration to Field Organizations will discuss a selection of case studies models for spatial configurations that the author has designed and developed from 1999 onwards. The chapter will demonstrate how to increasingly integrate design behaviours and experiential knowledge into computational design models. The chapter will progress from models aiming to solve shape constraints to models that generate associations between spatial, geometric and user intentions with participation from the designer as key mediator. A transition from space without people to people in space is attempted.

The majority of models were developed within the professional design research team called Computational Design Research group (CDR) of Aedas architects' R&D initiative, which the author co-founded with Peter Oborn in 2004. The author directed CDR from 2004-14 and has since moved the group to WoodsBagot architects where the group is called SUPERSPACE. A large number of models were developed during teaching and researching positions at various universities, by the author himself or with students. The majority of academic models were developed during the 9 years from 2001-9 as senior lecturer at the University of East London (UEL), teaching the Master of Computing and Design. The MSc was one of the modules taught at the research centre called Centre for Evolutionary Computing in Architecture (CECA) at UEL founded by Paul Coates and jointly directed by the author from 2002. CECA also functioned as consultancy in computational design to the architectural industry and pitched for public research funding of which one project will be discussed.

Other models include work from The Department of Architecture and Planning (DIAP) at the Politecnico di Milano in 2004, the MSc Adaptive Architecture and Computation at University College London (UCL) in 2007 and the chair for Emerging Technologies at the Technical University Munich (TUM) in 2011-12.

4.1 STRUCTURE

Four chapters will illustrate the increasing associations between observer, space, (meta-) heuristics, behaviours, and cognitive affordances in computational models of design. The four chapters discuss

5) Object Configurations

Models solving shape constraints based on specific computational heuristic concepts. All algorithms follow the bottom-up principle and generate self-organizing spatial configurations. The term of 'object' is used to point out that no human-centric constraints or objectives are involved. The algorithmic methods discussed provide the basic generative mechanisms.

[31] The title is a transformation of Stan Allen's essay title From object to Field: Field Conditions in Architecture and Urbanism (Allen 1997).

6) Mapping Spatial Associations

Models of spatial analysis and computational mapping techniques in relation to occupant behaviours and cognition. After having introduced generative self-organization methods, evaluation methods are discussed that provide measures for human-centric analysis of spatial configurations.

7) Synthetic Configurations

Models of design heuristics that generate spatial configurations based on spatial and occupation analysis. The first synthesis for *designing* spatial configurations with human-centric performances. Generative models of chapter five and analytical models of chapter six are integrated with different levels of autonomy, subject to the role of the observer.

8) Field Organizations

Meta-systems of design with open associations between models. The second synthesis with generalized design systems, leading to the Open Framework for Spatial Simulation (OFSS), developed by CDR over 10 years based on theoretical foundations introduced in chapter two.

Therefore the general structure follows a shift from object assemblies to human-centric field configurations.

While the USOM (User-centric Spatial Operations Model) represents the conceptual model as set out in the research objectives, the OFSS (Open Framework for Spatial Simulation) represents the active outcome or product.

4.1.1 Observer Agency

An inherent secondary structure within each chapter is implemented that refers to the role of the observer within a model. Three sections within a chapter identify the changing agency of the observer from

a) Remote observer

Agency of the observer is global, meaning he does not interfere directly with the processes of the model but sets out intentions, objectives and key performance indicators (KPIs) that a system should solve. The observer is remotely involved outside the main processing structure and does not interact with it. He guides the development either by encoding his intentions into the process by proxy such as agents or evaluates the process via explicit targets.

b) Situated observer

The observer introduces his agency directly by interacting with the algorithmic processes. His intentions, objectives and KPIs do not always have to be explicit but can be implicitly guiding the model towards some satisfactory state. The hierarchy between model and observer is softened. The situated observer is responsible for two roles: as *designer* he mediates the algorithmic heuristics and performance analysis processes with his architectural intentions; and as the agent of the *occupant* he mediates the perceptions and requirements of the user.

c) Learning observer

The observer watches the model associating spatial and cognitive categories and in the process proposes new categories that can represent feature classes, correlations between spatial properties or spatial typologies. The observer learns from the model. Most algorithms in these sections are self-organizing neural networks. (Chapter Five does not introduce the Learning Observer yet).

This secondary structure reflects the shift of control from the observer as global actor to an embedded local actor who is situated in an equally autonomous system. This shift is particularly relevant in the context of live design projects where design systems need quick assembly, trying to avoid tautological results.

4.1.2 Case Studies hierarchy

All sections in each chapter are introducing secondary case studies first where single aspects support the discussion. Sections end with primary case studies that best represent the concept under discussion and will be described in more depth. The order of the discussion is not related to chronological development of concepts or models.

Models in discussion often form part of a larger design system or research project but will be demonstrated in isolation. In chapter eight, section three – the final case study section – the relationship between some case study models that form larger systems as instances of the OFSS will be illustrated. Hence, models in chapter five to seven are not meant to solve a complex design system but represents a node in the system that facilitates human-centric spatial computation when synthesized into a design simulation framework.

Why is this structure important?

Designing systems – as opposed to academic simulations – require aspects of each discussed model and observer typology. The eventual OFSS encapsulates the potential to associate any of the meta-heuristics with spatial analysis, cognitive mapping and interaction to form the most *state-of-the–art* computational design system for user-centric spatial configurations in practice.

Maybe by chance, the structure approximates the historic development of artificial design systems, first from cybernetic production, to human analysis and eventually integration. However, if Bill Hillier's Space Syntax theory could be used as a reflection of this development (first a generative syntax, then user-centric spatial analysis), neither academia nor industry have managed to close the loop. This closure is attempted and demonstrated here.

4.2 TECHNOLOGY

Models produced by the author during his master of science in architecture (MSc) from 1999 to 2001 were written in C in the integrated development environment (IDE) Microsoft Developer Studio (MSDEV), which became Visual Studio. Compiled AutoCAD object libraries called AutoCAD Development System (ADS) were used to build executables to run from the command line in AutoCAD R14.

Since ADS was discontinued for AutoCAD from R15, models developed for and with students at CECA from 2001 to 2009, Technical University Vienna from 2005 to 2008 and the Politecnico di Milano in 2004 were mostly written in Microsoft Visual Basic for Applications (VBA) using application programming interfaces (API) of AutoCAD R15-23 and Bentley's Microstation V7.

Models written with and for students at the University College London in 2007 to 2008 and Technical University Munich in 2011-12 were written in Processing, which is a Java based programming language and IDE.

Models developed by CDR were written in different programming languages. From 2004 to 2009 VBA for AutoCAD R20-24 was used. In 2007 pilots were conducted through the then new Rhino 4 SDK for C# using the Microsoft Visual Studio IDE, Processing 1.0 and Java with OpenGL. Eventually the Java language was adopted utilizing OpenGL libraries for geometry and interfaces, developable with the Eclipse IDE. Since 2012 some projects have also been conducted in C# inside the Grasshopper API for Rhino 4 and the Python programming language with APIs of Rhino 5.

5 OBJECT CONFIGURATIONS | ALGORITHMIC GENERATION - THE MECHANISM

Object configurations relate to the solving of combinatorial problems. A series of explicitly shaped geometric elements is prepared to be combined into a larger object, which is not driven by occupational performances. In an architectural and spatial context, this has a long tradition of either layout automation or massing aggregation (see Kalay (2004) for a general overview). The models in this chapter are evaluated for explicit externally set targets – design drivers – which can be evaluated as functions but are not simulating user behaviours. The targets therefore do not reflect occupational patterns, user heuristics or cognitive performances but simply quantities that must be complied. All models are based on some computational meta-heuristic, which helps to search a complex design space, difficult to explore manually. Therefore, the models of chapter five reflect the research aims of the New Epistemologist who sought to find architectural expressions from algorithmic logic.

Robin Liggett identified three representations of space for layout automation problems (Liggett 2000): one-to-one assignments, which are discrete allocation problems between a single activity (room programme) and a single location without consideration of geometry or shapes of areas. Secondly, area representation problems or many-to-one assignments where the activities can be allocated to different areas, in other words an activity can take many shapes (this should really be called the 'one-to-many assignment'). And thirdly, the area-and-shape problem where both activities and the areas can be shaped and allocated in different ways as in floor plan layouts or space planning with physical geometry considerations. Another key distinction for her classification of layout automation models was the solution approach, which she also categorized into three types: single-criterion optimization, graph-based adjacency or topology and multi-criteria evaluation for feasibility of schematic layout design stages.

The models shown here fall mainly into the category of area-and-shape with multi-criteria evaluation for schematic design that Liggett (2000) still considered in 2000 the most difficult and rare in academic research. In fact, academia has practically abandoned space planning computation as it proved difficult to automate all possible constraints and heuristics. Liggett herself pointed out that most of those approaches are academic and that the area-and-shape problem, i.e. the space planning design approach for architecture, is the least consolidated in research. Amongst the reasons why industry has not picked up academic models, she lists "*a [lack of] modern interactive interface*" and "*support for iterative design process*" (Liggett 2000, p. 212). The models that have been developed by the CDR group are mostly built with those properties in mind because they were aimed for 'live use' by architects rather than automation of knowledge and heuristics.

Models of section 5.1 constitutes the most direct mapping of spatial configuration through computation as the solution states have to attain explicit numerical targets, which can be measured. These models fall generally into the category of optimization problems where the search process aims to find combinations that increase the value or decrease the cost of an evaluation function.

The most obvious way to attempt to solve layout problems would be by solving parametric dependencies through exhaustive enumeration. But this is infeasible as the complexity of a design problem and the size of the solution space increase exponentially with each element that needs solving for. In mathematics this type of problem is called NP-complete (nondeterministic polynomial time problems as the complexity of the problem cannot be determined to be solved in a feasible timeframe. When such problems occur, approximation or meta-heuristics are valuable alternatives to search for a solution space that can be reached in feasible time. Approximation and meta-heuristics on the other hand, are not guaranteed to provide the best possible solution but consistently provide a nearly optimal solution. Meta-heuristics provide a search path that offers a method to approach the problem. This search-path can sometimes be loosely identified in the heuristics of a design method, which will be mentioned for each model if applicable.

5.1 OBSERVER COMPOSITIONS | CONSTRAINTS SATISFACTION

Most combinatorial models have explicit cost functions. For space planning these represent the evaluation of design drivers set as targets at the beginning of the design process. If design drivers and their targets are given such as achieving the maximum amount of flats within an envelope, then the evaluation function is generally coded into the model, replacing the observer as a judge of performance. This also means that a clear division of labour exists between model and designer-user, meaning the designer watches the model solve rather than participates in its search. The model is employed as a tool for efficiency and speed within existing design workflows instead of introducing new thinking about the design process.

In this context of solving for specific and explicit design drivers without interference by the designer, the approximation approach chosen most often by the author represents an evolutionary search method called the *genetic algorithm* (GA, see 3.1). The fitness functions evaluating the design drivers vary by brief but generally when hardcoded they can be regarded as the design drivers that the observer applies to decide whether to keep or dismiss a state or phenotype.

GAs lend themselves well to combinatorial design processes with specific shape elements because of the division of the generative system and the evaluation function. The generative system encodes data of the embryology into chromosomes consisting of gene strings that can encode geometric structures, including parametric dependencies. The evaluation in the form of *fitness* functions can represent any observer-designer criteria and in fact judges from outside the generative process the state of a spatial configuration rather than the individual shapes encoded. If therefore, the designer is not searching for new morphologies for a spatial configuration, the fitness function serves as encoded agency for the observer to quickly find assemblies of given geometries, depending on the complexity of the evaluation criteria.

5.1.1 Single Criterion Optimization

The simplest forms of evaluating an observer's design driver are single criteria optimization GAs. For single criteria it actually makes sense to talk of *optimization* because the targets of drivers cannot conflict, which usually leads to trade-offs rather than optimal targets for all drivers. The curriculum of the programming architecture module of the MSc C&D at UEL, taught by the author between 2001 and 2009, included experiments with single criteria GAs for spatial configurations. Basic GA codes prepared by the author in VBA for AutoCAD were introduced to the students who then elaborated the embryology or fitness functions.

CECA TEACHING GA

The basic model that the author provided to students at CECA from 2004-8 evaluates for a single criterion that represents a version of a plot ratio used in the profession to evaluate the density of built-up volume on a plot area, which in turn is a measure of investment efficiency. The function simple calculates

individual.fitness = boundingbox (of plot) / volume

producing a coefficient that expresses the amount of volume per area. *Roulette wheel* selection is used as selection function as it does not converge too fast onto a limited expression of chromosomes (*genetic drift*) because not always the best individuals of a generation are picked; in fact, generations can temporarily become weaker since they include weaker genetic material (Fig79). The genotype here encodes the control points of five polylines and ratios of rectangles (length and width) that are extruded along the polyline paths. The extruded volumes are unioned into a single volume, which emulates a kind of deconstructivist building geometry. The single evaluation criterion could then be applied to the resulting volume and other types of ratios such as length over volume. Playing with those simple ratios allowed students to watch the 'blind watchmaker' aka algorithm generate forms constrained by simple selection that otherwise would semantically be perceived as 'slender', 'rugged, 'tall', 'bulging' etc.

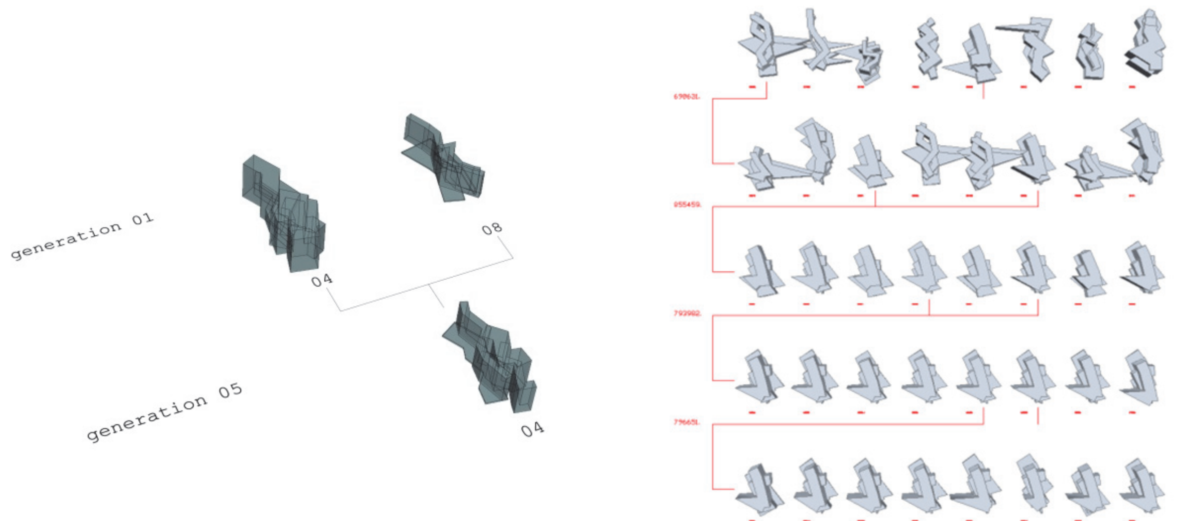


Fig79. *GA using the CECA basic code provided by author with plot density evaluation; (left) selected parents and phenotype; and (right) five generations where the third generation has a lower average fitness than the previous one*

SPACE PACKING

Another such single criterion GA for area efficiency optimization was applied to the Khalifa-bin-Zayed al Nehayan competition. The client tendered a competition for a mixed-use development in Abu Dhabi in 2008. CDR developed three models for the cladding and roof plan of the retail plinth and the packing of residential apartments into a given cylindrical envelope that rose above the plinth. The combinatorial problem of how to pack a desired mix of apartment volumes into an exact envelope lent itself ideally to an evolutionary search. In fact, contrary to the previous student exercise, it appeared the only possible way to find a solution in the short timespan dictated by the deadline.

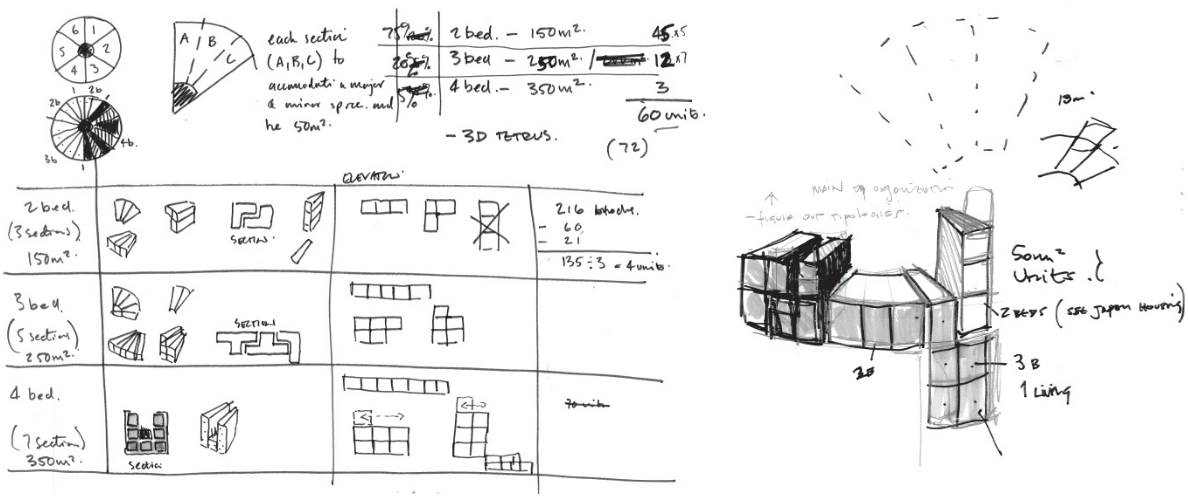


***Fig80**. Khalifa-bin-Zayed residential units were categorized by three types of apartments based on number of spaces and their distribution across vertical floors, resulting in a matrix of permissible packing tiles in 3D*

Åsmund Gamlesæter's development for the residential tower comprised three stages: geometry development to explore and align with concept design, optimization mechanism for general volumetric packing and finally a constrained optimization to comply with the final apartment mix (Helme et al 2014). The first application still assumed a Tetris-like packing to be a 3D volumetric study where an apartment layout could vary in elevation as well as depth on a radial grid. But the size of search space would have become so large that neither the architects developing layouts nor the model searching for packing solutions would have arrived at a well-defined result within the given time. The second application therefore reduced the search space to apartment layouts that vary only in 2 dimensions, albeit in elevation while still no final set of apartment types were provided. The problem is similar to tiling with polyomino puzzles, which is an old combinatorial geometric game. Where polyomino tilings are usually done with only one size of tile (number of squares) like Tetris, which has four squares that pack edgewise into different shapes without holes, three types of apartments with different possible area combination were tested representing a polyomino with many different-sided tiles. The second application used an evolutionary search based on a standard GA and attempted to pack the envelope while concurrently producing a set of apartment layouts. To do so, each individual subdivided the cylindrical radial grid by placing walls randomly checking how many apartment areas could be fitted. Yet this approach also proved too time consuming and incompatible with desired layouts.

Eventually, a catalogue of apartments was manually laid-out compliant with the requirements for two, three and four bedroom apartments. Two bedrooms equalled variations of a tromino (3 squares) x 45 apartments, three bedrooms equalled variations of pentomino (5 squares) x 12 apartments and four bedrooms equalled variations of heptomino (7 squares) x 3 apartments. The genotype only comprised 4 binary positions for (1) apartment type, (2) mirroring, (3) vertical and (4) horizontal grid location, resulting however in a total pool of apartment configuration types of

2 bed x 3 squares (150 sm)	:	45 x 2 variations	= 90
3 bed x 5 squares (250 sm)	:	12 x 3 variations	= 36
4 bed x 7 squares (350 sm)	:	3 x 9 variations	= 27
10800 sm	:	60 target apartments / 153 available	

Each genotype then carries

*6048 permutations = 14 variations x 2 (mirror) x 216 (18 horizontal * 12 vertical)*

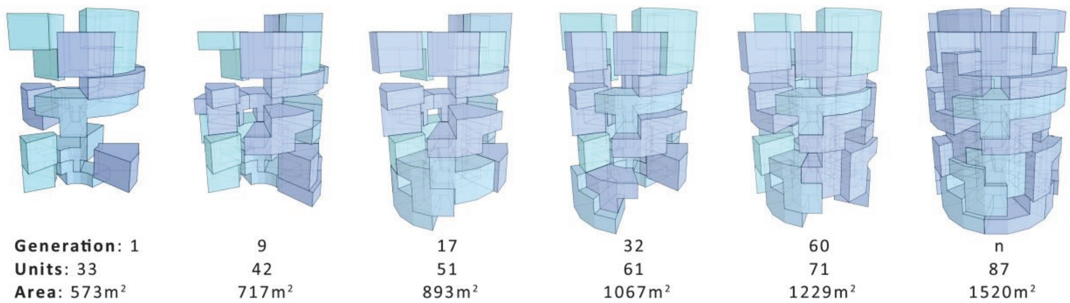


Fig81. *The second GA for Khalifa-bin-Zayed, increased the density of apartments within the cylindrical tower, leaving empty units between neighbours for communal areas*

This last of the three applications of the combinatorial packing model was represented through an unfolded elevation, i.e. a planar puzzle like the polyomino. Each generation had 200 individuals, with each individual attempting to place all 60 apartments into the 216 grid cells. The fitness function evaluated the amount of overlap of grid cells in each individual that was composed of 18 horizontal by 12 vertical cells, dimensioned according to the physical space allocated to each apartment unit:

individual.fitness = no. of total cell overlaps

A roulette wheel selection extracted the fitter 100 phenotypes to breed the next generation. Due to deadline, no optimization of the fitness evaluation and selection procedure was attempted, resulting in runs that got stuck in local maxima. The model had to be run repetitively before producing completely packed solutions. Eventually, a solution was picked where the larger 3 apartments were allocated further up for better views and light.

Apart from the properties of efficiency and speed to assemble a complex combinatorial geometric puzzle, the model allowed the design team to experiment with non-uniform layouts across floors. Floors in towers with a uniform envelope are generally repeated.

The Khalifa-bin-Zahed mixed use competition provides insight into the limitations of a competition setting where a multitude of teams collaborate and synchronize intermediate output within a short period of time. Still, the computational models provided the design team with validation for their assumptions about the apartment layouts and numbers, and freedom to design non-uniform floor configurations.

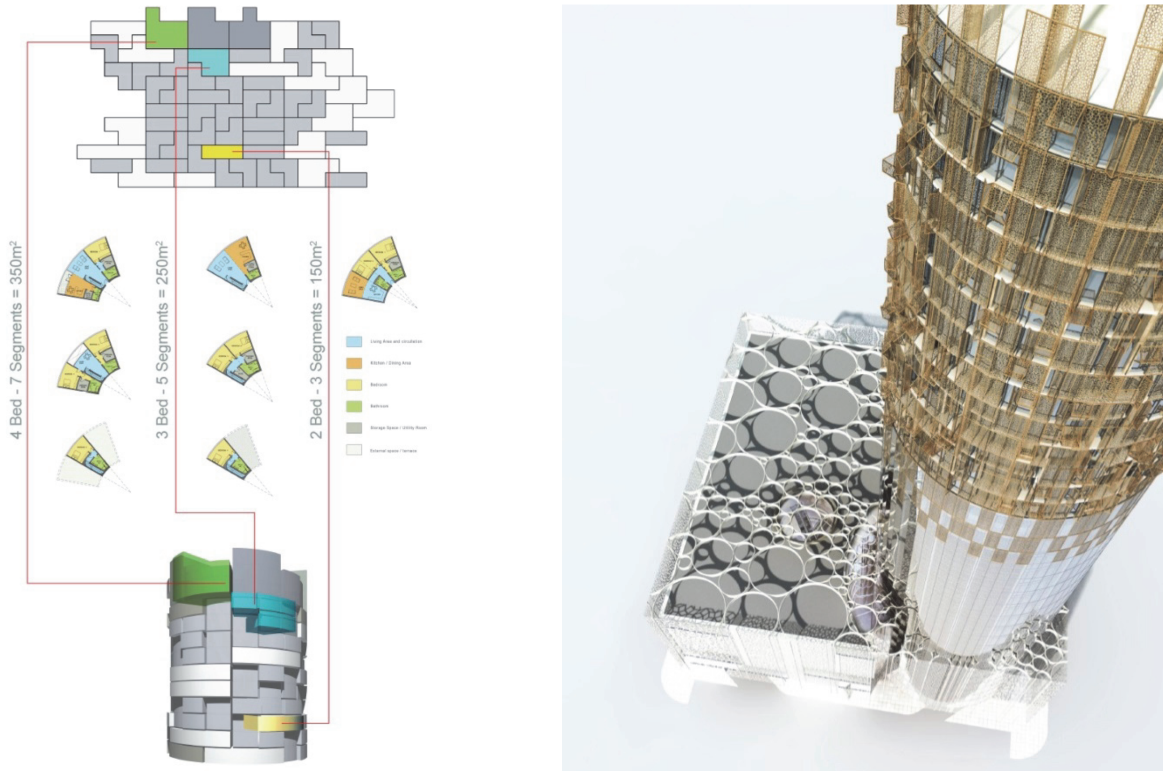


Fig82*. The third GA for Khalifa-bin-Zayed unfolded the elevation and packed the permissible apartment layouts with level constraints (top left); the resulting assembly was rolled back into the cylindrical tower (right)*

TOPOLOGICAL BREEDING

On an urban scale, single criteria can represent many parameters. In 2008 Aedas was commissioned to conduct a study for a mixed-use masterplan for the infrastructure and housing arm of Sahara India Pariwar conglomerate who develop ‘townships’ with the intention to network them into ‘mega townships’. CDR in turn developed a small prototype as a proof of concept to generate quick capacity scenarios, given the site perimeter and development quantum. A case study site provided only a blank canvas with some access points into the site. Generally, the model was intended to apply to any blank site.

Given the low amount of constraints and generic character, it was decided to create a two-stage model that would first generate a partition of the site into roughly even sized plots, which could be filled with the quantum evenly in a second step. CDR member Pablo Miranda developed the generative site partitioning model based on a steady-state genetic algorithm.

The site partitioning splits the site into a series of plots with approximately even areas for development (requested at the time by collaborating design team assuming that this is a good condition). The input to this model consists only of a site perimeter polyline and locations that the partitions need to run through. The partition edges are considered to be like movement axes while locations are mostly placed on the perimeter as site access points but also as desired public locations within the site where accessibility needs to be guaranteed. The genotype consists only of the direction vectors of the axes that run through the access point. Phenotypes only decode area polygons that result from the axes partitioning the site. The genetic algorithm searches for individuals with direction vectors that split the site into areas that are as even as possible. Thus, the single criterion fitness encodes the standard deviation between the areas of an individual

individual.fitness = sqr(sum(individual.areas[i]-mean))/num_polygons)

The selection process for choosing individuals for breeding is a simple rank binary tournament. Two randomly picked individuals in a generation are evaluated for their fitness. The less fit individual, i.e. with the higher standard deviation value, is crossed-over with the fitter individual (including some mutation probability). If the new individual has a better fitness than the previous loser, it replaces the loser in the population. Breeding and replacing only selected a few, or in this case one individual at a time, is called steady-state evolution, where only improved individuals are replaced rather than a whole population.

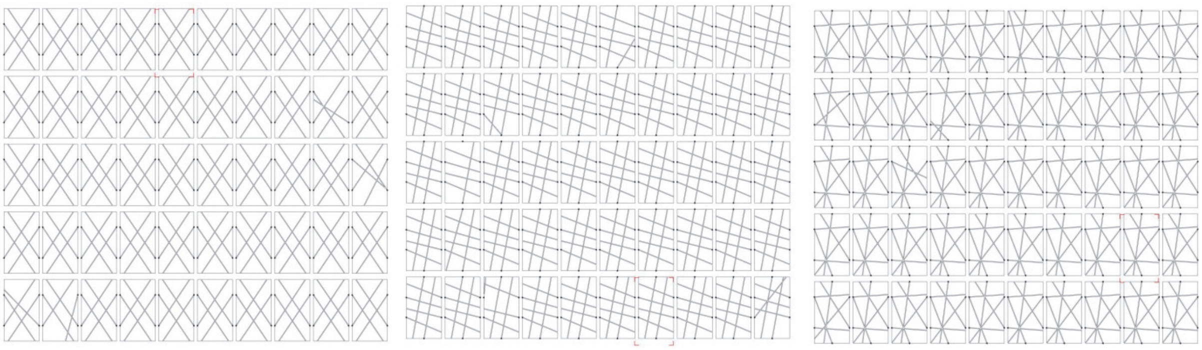


Fig83. *The Sahara Township steady-state site partitioning model showing three screenshots of settled configurations with differently seeded site access points; instead of linear connections, the axes intersect to provide high accessibility and even area plots*

Selecting and breeding for minimum deviation of areas does not explicitly set axis alignment targets or topological relations. But again, a semantic and structural side effect of this selection criterion by equal areas is a derived logic of topological distribution of intersections between axes. The network of axes produced automatically seems to converge towards evenly spaced intersections and often symmetry. An emergent property that was intended however and embodies a network accessibility property, i.e. the more evenly the intersections between axes are distributed the better the accessibility between the plots on site. This is a property demonstrated by Bill Hillier in the Fundamental City (1996, p271ff) that can be observed but was not hardcoded into the system. The resulting network of axes partitioning the site into approximately even plot areas is exported by plot polygons and in the next step can be populated with the development quantum. The quantum and its proportions of mix are evenly distributed into each plot so that a density per plot can be visually grasped and refined.

Models presented in this section appear to emulate the heuristics of a designer through a computational meta-heuristic, namely the relatively simple genetic algorithm. Visualization plays a strong role in the models to make the computational process accessible and intuitive. The first basic area optimization shows the genealogy and therefore allows the designer to gain insights into the selection process, creating a consensus by visual inspection. The second combinatorial optimization only ever shows the best individual in a generation concentrating instead on the visualization of the re-arrangement of the apartment configurations. In other words it makes the hidden process of crossing-over, mutation and selection palpable, just as a designer would test puzzle pieces and replace them. Especially, the last application rendered two planes that show the packing apartment modules on the lower plane and the overlapping apartment modules on the upper plane as if the invisible designer-algorithm was 'holding' up the pieces that do not fit yet contemplating the current solution before replacing pieces. Also the last axis network and plot area model made the testing and choosing of the algorithm obvious, so that the observer-designer feels compelled to interfere or communicate with the machine as if it was a collaborative designer.

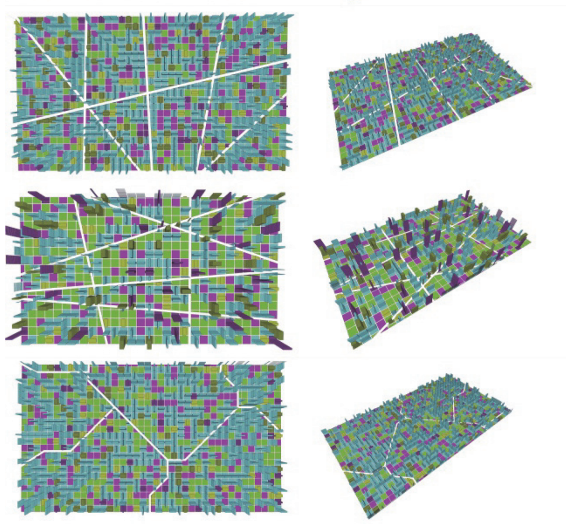


Fig84. *The Sahara Township steady-state site partitioning model showing three screenshots of the population application that inserts the development quantum into the carved out plots; each plot attempts to accommodate the same mix of uses represented by the colours*

Clearly, the computational meta-heuristic does not directly mimic design heuristics. Especially, in the first two models the designer would replace individual apartment modules instead of an entire arrangement. But in all models the visualization comes very close to evoking the designer's behaviour and thus appears to simulate the reasoning behind his heuristic.

5.1.2 Multi-Criteria Optimization

When a designer is searching for solutions with more than one target criterion in the cost function, it becomes less obvious what such a target state would look like. This means that the genotype probably does not only contain information about shapes and dimensions but also about relational constraints such as topology or external drivers like climatic variables. As a consequence one should not talk about optimization at all since the diverse criteria that the fitness function has to aggregate, might be conflicting. Rarely do the fitnesses of two (or more) criteria symbiotically enforce each other but are optimal at each other's expense. Hence, the selection of individuals needs to be weighted so that no single criterion dominates the fitness calculation, which inevitably leads to best compromises (usually called *trade-offs*) rather than optimal criteria fitnesses. Additionally, explicit visual and performative design intentions are more difficult to represent as evaluation criteria, because the weightings between possibly conflicting criteria performances do not provide a clear picture of the configuration states. The observer will have to hand over more autonomy to the generative process, especially here the fitness and selection functions.

INTEGRATED MULTI-STAGE OPTIMIZATION

Starting with a simple example of such a compromise again, a demo project to generate tower-like architectures was conducted by the author at CDR in 2006, called Faulty Towers. The overall intention of the project was to demonstrate that diverse project stages in a workflow could be integrated into a generative model through techniques of artificial intelligence and life. This 'digital chain' represented the design of a multi-storey building, comprising

a. the allocation of the area programme
b. translating it into a geometric diagram representing the actual dimensions of space
c. optimizing floorplates and envelopes

A. encoded the room schedule and adjacency matrix in a Cellular Automaton and allocated voxels (3D pixels) according to areas, their adjacencies and proximity to external skin for daylight and temperature.
B. translated the resulting topological diagram into a constrained morphology as an implicit surface, translating Paul Bourke's Marching Cubes algorithm (1994) from C++ to VB. This prototype is analysed for solar exposure performance and cladding dimensions. Then it is encoded into the genotype for participatory evolution.
C. used a standard genetic algorithm to allow architects to continue to evolve the morphology based on floor-to-wall ratio and envelope performance.

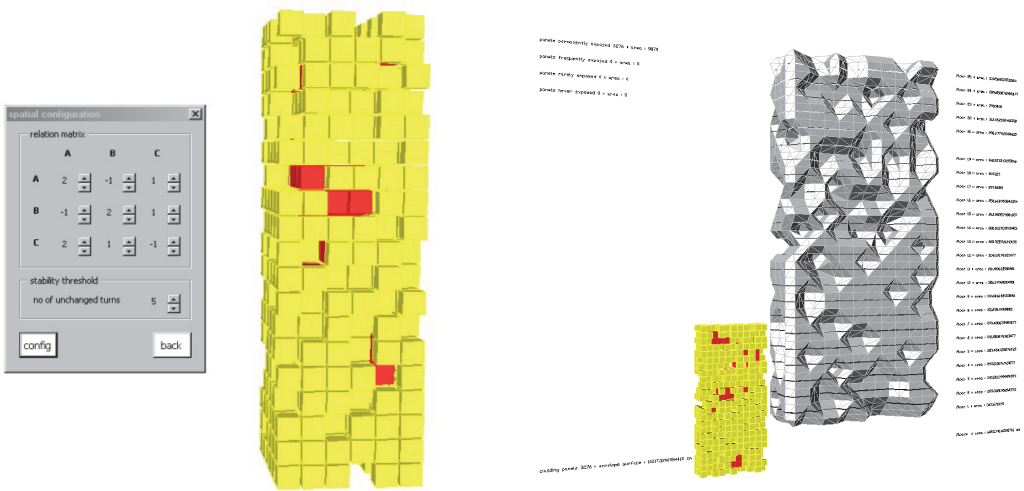


Fig85. *Faulty Towers Stage A Cellular Automaton allocating programme across the topologically partitioned volume (left) and its stage B geometric translation (right)*

The innovation of stage C in this process was the fitness and selection evaluation procedure: a hybrid between natural and artificial selection. The visual interface presented all 8 individuals of a generation to the user who would have to pick an individual to breed the next generation with. Similar to natural selection, no selective breeding takes place but instead the observer or a group of observers provides the contextual undirected selection. The morphology could be inspected on screen, the envelope hidden to review the floor plates and areas for obvious (dis-)advantages. The morphology is controlled via the *isolevel* parameter in the MCA that regulates the tightness of the surface to the topology. The user therefore sub-consciously selects individual traits based on isolevels, which are encoded in the genotype.

Fig86. *Faulty Towers Stage B envelope evaluation by skin fitting and solar analysis*

The artificial selection evaluates only two criteria or traits, namely total gross floor area and envelope surface area. The ratio between the two is commonly known as floor-to-wall ratio in the industry. But to select for a good ratio within a generation is not trivial despite there being only two criteria because their optima conflict: a good floor area produces large envelope surfaces and vice versa, good envelope surface areas produce small floor areas. In such a case, if the fitness was calculated merely as a ratio between the values of the two criteria, meaningless architecture would result because the best ratio would strive to produce sphere-like blobs.
Since it is counter-productive to simply create a value-based ratio or co-efficient between the two criteria, an abstraction is required. To calculate the fitness of each individual, the two criteria are ranked separately *but* the index of the lists need to reflect the performance of the criteria. That means that floor-area is ranked from index 1 = small area to index 8 = large area, and for envelope-surface index 1 = large area and index 8 = small area. The two ranks are now multiplied creating a composite rank product for each individual that reflects a compromise between the conflicting criteria. A rank selection is conducted that establishes the best compromise in the generation. If more than one individual achieves the highest rank product (which can happen frequently because the rank lists are fairly short), a tournament selection follows where the individual with largest floor area prevails. The individual selected naturally by the observer for implicit traits and the individual selected artificially by the algorithm for explicit traits then breed a new generation.

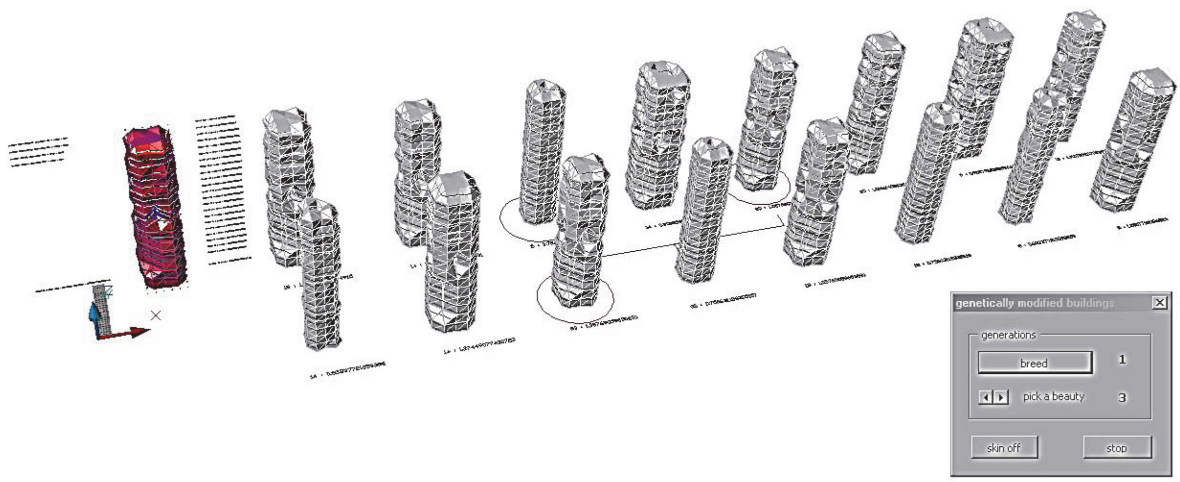


Fig87. *Faulty Towers Stage C optimizing floorplates and envelops using a hybrid natural-with-artificial selection genetic algorithm; one circle in the first generation shows the selection by the algorithm, the other circle the observer's choice through the small GUI element*

The rank product has some interesting characteristics. First, the abstraction of metric values into index values normalizes the two criteria dimensions that vary significantly in their maximum ranges. Secondly, it avoids the domination of a single criterion which would produce meaningless phenotypes but rewards compromise. For example

individual.fitness = floor index 8 (large) x envelope index 1 (large) = 8

producing a deep (fat) volume, while

individual.fitness = floor index 5 (med-large) x envelope surface 5 (med-large) = 25

produces a well-proportioned volume, and

individual.fitness = floor index 8 (large) x envelope surface 8(small) = 64

produces a tall and slender volume, not too deep for natural lighting.

The most important aspect of this rank product representation is however the relation to the observer. In contrast to the previous single-criterion optimization, the designer cannot simply encode his intentions as observed or represented numerically outside the system, here the floor-to-wall ratio. Instead, the observer needs to see the system perspective which allows transparency of the relational abstraction between dimensions. He can then observe and participate in the process and realize that the selection by systemic logic, i.e. not aiming for short-term gains, achieves sustainable and better results; especially when breeding with implicit observer selected morphologies, which often do not score highest fitness values. The algorithm and fitness function represent various stakeholders who construct a consensus. The observer participates in this consensual construction rather than imposing his specific target on the system.

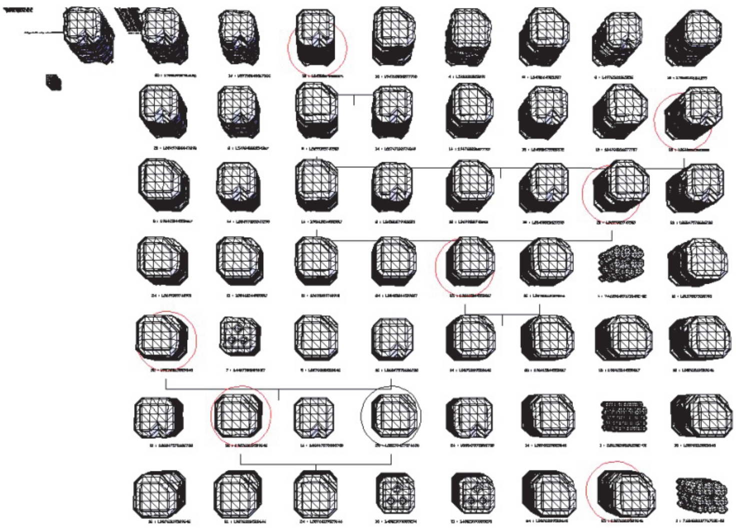


Fig88*. Faulty Towers family tree at the eighth generation of an observer-algorithm consensual construction*

PARETO MASTERPLANNING

In contemporary commercial masterplanning, the brief and the methodology for designing the outline massing and mix tends to be an area and density optimization problem, attempting to achieve an efficient balance of large data sets. Masterplanning data for spatial planning constitute the development schedules, area proportions, proximities and regulatory controls, all in relation to revenue and costs. In standard practice, the land-use types, their areas and the proximities to site conditions are set in the area schedule. Further information about density levels (plot ratios) and mix are provided that in general constitute a framework for massing. Like the Khalifa-bin-Zayed competition, there is a catalogue of space types and permitted layouts that need assembling. Just that in the case of a masterplan there are many more constraints and infinitely more combinatorial possibilities, i.e. the search space is NP complete. Not only is it impossible to find good solution in finite time but there may also be many permissible solutions depending on stakeholders' bias, making it a typical wicked problem.

In 2005, CDR was commissioned by the 4M Group[32] to collaborate on a masterplan in Pristina, Kosovo. The mixed-use masterplan called ENK Complex demanded 153000 m^2 of retail, offices, hotel and residential areas. The masterplan was designed by 4M and the elevations by CDR. The ENK Complex was used as a vehicle to test the development of a complex multi-criteria optimization model for masterplanning with a sponsored student, Edward Finucane.

The model developed by Finucane under supervision by the author, represents a proof-of-concept models that integrated many steps of the design process to generate options of massing diagrams on site from the area schedule. A genetic algorithm was used with a complex selection function based on the Pareto Optimization method, which will be described below (Finucane et al. 2006). The embryology consists of a 2D grid for the site and 3D grid of cells for the massing that encodes

[32] http://www.4mgroup.co.uk, accessed 05.08.2012

a. specification of building footprints by delimiting the public space through an automated circulation diagram on a 2D grid
b. allocation of land-uses within the footprints (developable plots)
c. translation of massing volumes into a 3D grid

A. An Ant-Colony algorithm provides the partitioning meta-heuristic for the circulation network between access points including entrance/exits to and from the site perimeter and desired public spaces within the site. Ant-Colony algorithms are regarded as an optimization method (ACO) due to their property of finding the network of shortest routes from a set of source and sink points (nest and food sources). Site perimeter access points represent the nest points and public space access points constitute the food locations. The simulated foraging ants create a network of circulation routes between the locations by using pheromone trails to find the quickest paths between nest and food points. The pheromone trails are laid on the way back from a found food point location, evaporating if not reinforced by other or the same ant (Panait and Luke 2004). The routes are offset on a grid according to the width of street and path dimensions specified by Kosovan regulation. The resulting circulation area must not exceed 30% of the plot area as 70% is dedicated for development.

B. The cells of the 2D grid include a Boolean state that indicates a built/unbuilt location. A number of grid cells with the built-state are randomly seeded with a land-use that sets their use-state. A reaction-diffusion cellular automaton diffuses the land-uses across the building footprints. Where two states meet, they create a boundary. The land-use diffusion is constrained by plan depth and thus cannot be placed everywhere within the grid. Each cell also carries information about the height of the building by number of floors per land-use. Thus, a 2D map of land-use distribution is generated providing the information for the 3D volumetric grid.

C. A 3D grid is produced on the basis of the floor number value of the 2D cells. Again, land-use seeds are randomly place into the 3D grid, which however are constrained by the area schedule that regulates height restrictions per use. A passive diffusion entails where non-attributed cells read neighbour states vertically and horizontally to set their own use-state (an 'infected' rather than 'infecting' absorption). When all cells have attained a use-state, the 3D grid is translated geometrically into floors, resulting in an area massing.

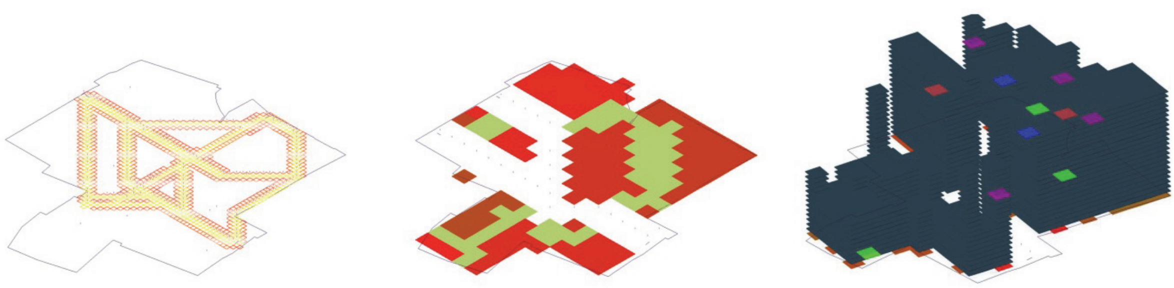


Fig89*. The embryology of the Pareto masterplanning model using in stage A an ant-colony algorithm to generate a circulation diagram (left), which is diffused into the carved out plots using a cellular automaton in stage B (middle) and in stage C arrayed vertically and land-use cells are inserted randomly based on quantum (right)*

Thus, the binary gene string (chromosome) of the embryology contains alleles for

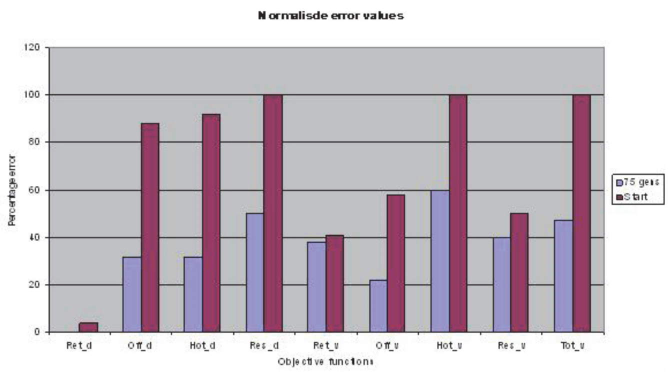
1. positions of food sources (accessible open public spaces for building footprint development)
2. number, positions and max heights of seeds inside the footprints
3. number and positions for seeds of volumetric grid
4. land-use state

6 cross-over points are given for the cross-over function of chromosomes in order to provide a good mix. The population size varied in experiments around 40 individuals running for 75-100 generations.

The constraints with which the phenotypes were generated consist of

	Retail	Offices	Hotel	Residential	Total
volume (m^2)	39000	30000	7500	70000	153000
floor heights	-1 to 3	2 to max	6 to max	2 to max	
plan depth (m)	min 15/ max 45	min 13/ max 21	min 8/ max 20	min 8/ max 21	
built/ unbuilt (%)					30/70

All but the floor heights and total area were used as evaluation criteria, giving 9 target values for the cost functions. As shown in previous projects, the higher the number of criteria, the more carefully the fitness must be calculated. Additionally, with such a high number of criteria, the selection process becomes complicated. The aim of developing a masterplan is to ideally distribute the land-uses in the development quantum such that no allocation of one disadvantages another, meaning that all land-uses must find a compromise distribution where the minimum required constraints of all land-uses are fulfilled. This type of optimum configuration based on trading off criteria has been developed by the Italian economist Vilfredo Pareto in the 1920s and is called Pareto Optimality. In general, this type of equilibrium optimality means that no single criterion will potentially reach the desired full target value or even the state under which it would perform best, i.e. it might not reach an optimal state in itself. It also undermines the notion of being able to pre-set the targets of a good or bad final state. Many configurations might be equally good whose structure cannot be anticipated by an observer but result from the process.



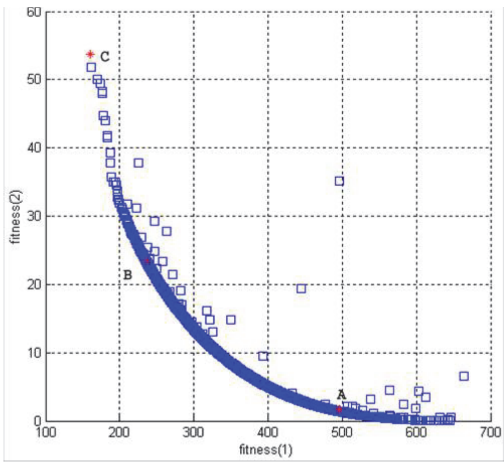


Fig90. *Error reduction of each KPI before (purple bars) and after (blue bars) at 75 generations (left); and the first Pareto front (right) showing archived individuals distributed by weighting of criteria across the front*

Pareto Optimization was first proposed for genetic algorithms by Goldberg (Goldberg 1989) and aims at evaluating the fitness of individuals by the number of criteria that are not dominated by other individuals. Each criterion calculated the error margin towards the desired targets

individual.fitness.criterion(i) = criterion(i).current_value – criterion(i).target_value

Then the criteria for each individual are compared and the fitness score is given by how many other individuals' criteria are dominating

If (individual(i).fitness.criterion(j) < individual(n).fitness.criterion(j))

Then (individual(i).fitness.dominated_criteria += 1)

All individuals with the same number of dominated criteria are assigned to a list called a 'Pareto front'. A number of fronts are created within which all individuals in a single front are equally dominated. The 'first front' contains all non-dominated individuals representing the best compromises found, the 'second' front those with one criterion dominated and so forth. The fronts are assembled over generations as each generation cannot produce many best compromises. The Non-dominated Sorting Genetic Algorithm (NSAG) employed here is based on a development by Srinivas and Deb (Srinivas and Deb 1995).

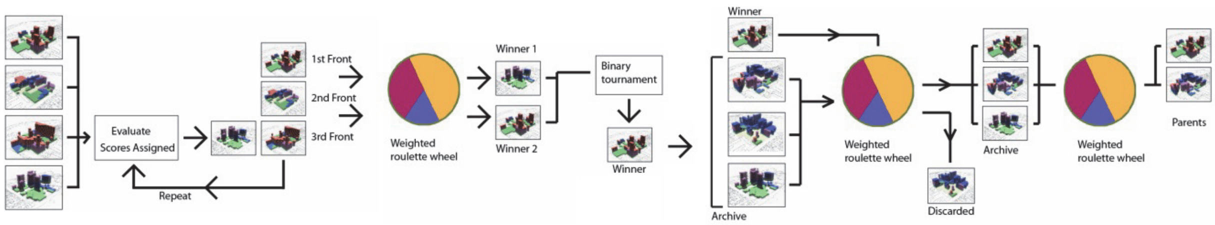


***Fig91**. The selection process based on the PESA algorithm (Corne et al. 2000) to extract individuals into archives that represent Pareto fronts*

The entailing selection function is based on the Pareto Envelope-based Selection Algorithm (PESA) (Corne et al. 2000), which creates an archive of best compromises over generations from the first Pareto fronts. PESA therefore works with two populations: the bred generation as internal and the collected archive as external population. The archive initially simply represents the first generation but is optimized over time by inputting non-dominated individuals from each generation. The individuals in each Pareto front are given a rank according to the amount of non-dominated criteria (i.e. equivalent to their front). A roulette wheel selection extracts two non-dominated individuals by higher probability of their rank, which in turn establish a winner through a weighted binary tournament selection by comparing their criteria. This winner is included in the archive, which holds a fixed number of individuals – here 20. To keep the number steady, a second roulette wheel selection identifies an individual from the archive that is discarded. The roulette wheel at this stage is weighted towards a 'niche' ranking. Each individual in the archive is given a niche value calculated from the distribution density. If an individual is situated in a crowded location in the archive, it is penalized by a decreased probability variable. The niche value is calculated by distance within a front between individuals. Crowded locations in a front are sources of genetic drift and are ideally controlled by the niche value to have less weight (Horn et al. 1994). A final and third roulette wheel selection chooses two winners to breed the next generation from before starting with the decoding of the phenotypes from the above described embryology again.

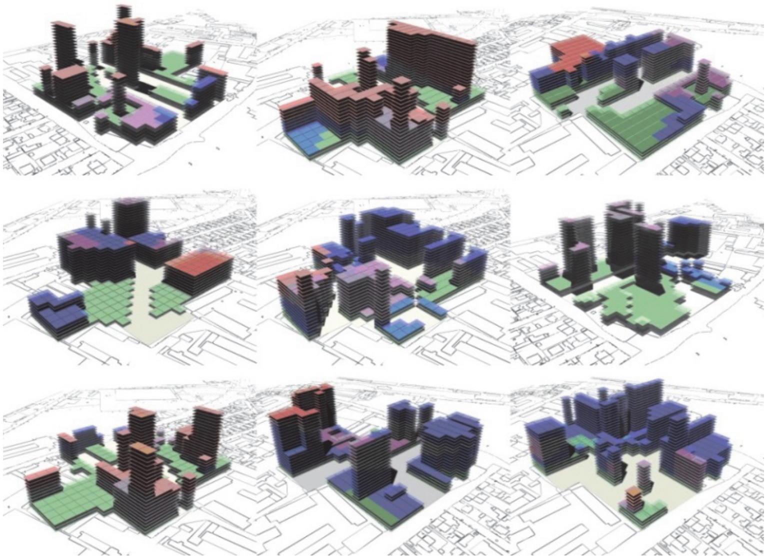


Fig92*. Nine massing optoins from the Pareto masterplannning model for the ENK Complex with colours indicating the land-use*

The results of this complicated process for masterplanning were encouraging. Within 100 generations, the error margins for all criteria decreased synchronously and some meaningful massing solutions could be presented that approximated the area, depth and plot ratio given in the development quantum. Since the process was built on a modularity of design steps, the model could be adjusted and used for the Smart Solutions for Spatial Planning (SSSP) project, which was developed with funding from the UrbanBUZZ grant in 2007 in collaboration with CDR. For SSSP, the ACO was removed and a more complex geometry partitioning algorithm based on a Lindenmayer system introduced (Derix 2012a). The phenotypes represented urban blocks based on an embryology including land-use mixes across several floors, density by height, depth of block by use and porosity of the block perimeter for access (Coates and Derix 2008). The Pareto urban block optimization in SSSP represented the last step within a larger urban design digital chain.

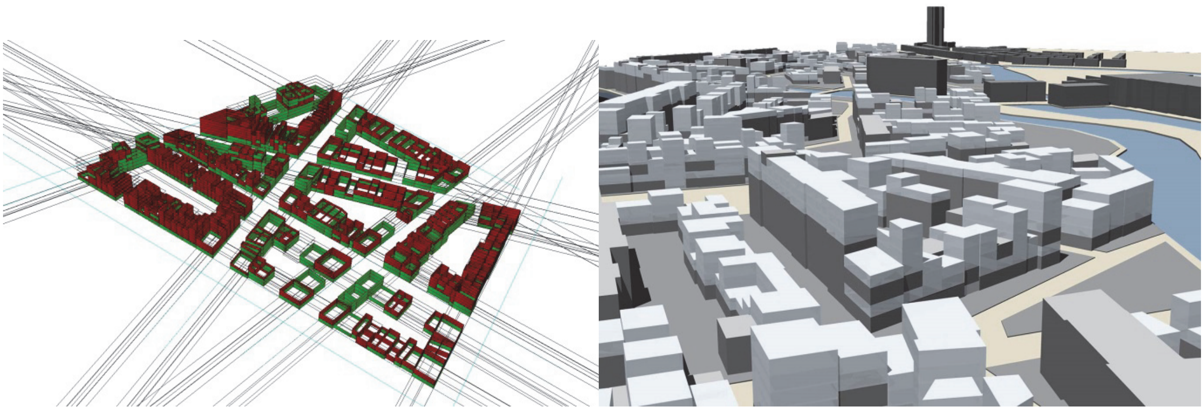


Fig93*. The Pareto masterplanning model using a Lindenmayer system as embryology to encode genotypes and decode phenotypes as urban block definitions for the SSSP project*

Faulty Towers and the ENK Complex projects represent early proof-of-concept models aimed to demonstrate the capacities of generative computational design with artificial life algorithms, yet not intended for routine design generation. This type of model integrates many design stages of the overall design process and therefore becomes too specific for application on other projects.

As opposed to the single-criterion optimization, the complexity of the multi-criteria optimization models make it difficult to visualize discrete steps of the processing mechanism that build identification between design and computational heuristics. However, the archiving and weighting methods that control the selection over generations in the ENK Complex project approximate the strategy of categorising and choosing design configurations in a design process. Rachel Cruise (2005) described this strategic selection process in her Dry Stone Waller project , using as an analogy a builder's heuristic by which he evaluates and categorizes stones during construction (rather than pre-classification) by creating piles of different qualities that correlate to known conditions in the building process that could occur.

Evolutionary algorithms as described in this section 5.1 work on the basis of populations. Complex multi-criteria optimization models employing complicated internal selection functions are difficult to align with live design processes because interference with the developmental process is difficult to design into the models' complex internal procedures that require completion before being decoded and visualized for interaction. The separation of generation and evaluation at the level of the spatial parts represents a modular structure that allows the observer to quickly adapt the targets of design drivers for the whole system but the generative process at the level of the local elementary relations is beyond his reach. While this *black box* approach might be intriguing, it is not very useful for a live design process. The next two sections discuss models that allow for more local agency of the observer in the configurations of spatial configurations and shapes.

5.2 PARTS ASSEMBLIES | TOPOLOGIC SEARCH

Space planning as discussed by Muther (2012) starts by compiling the areas required by the brief, specific to client and typology. As a minimum, this includes the accommodation schedule and adjacency matrix. At first a designer attempts to loosely solve the adjacency requirements, traditionally done by using either cut-out shapes or virtual shapes in CAD. He moves the place-holder shapes sized to the adequate areas from the schedule around one at a time until all proximity preferences between the rooms and their groups (if done hierarchically) have been resolved in such a way that the configuration complies explicitly with the specific brief and implicitly with the experience of the designer. When the general topology has been resolved with some simplified geometry, a more detailed stage follows to refine the geometries that facilitate specific uses, timetabling and site conditions. Both stages are more concerned with resolving local relations and geometries than global form. Despite the visual culture of architecture and the common obsession with iconic appearance, the global form must comply with the internal constraints first. Massing and envelope are also a result of internal processes if the parts of the system, i.e. spatial units, are locally responding to their situation (climatic, social, topographic etc). Therefore, the state space is topologically given but morphologically unknown.

OBSERVER-AGENCY

A designer who chooses to employ a computational model to generate spatial configurations as topological diagrams of morphologies, needs to understand the rules of local assembly as generator of global form. While evaluating the global form emerging from local assemblies, the designer mainly applies brief constraints and his empirical knowledge to geometric spatial units, i.e. specifies the local performances. A spatial unit is composed of the simplified abstracted geometric shape (*geometry*), its relational constraints (*topology*) such as adjacencies to other spaces within its operational context (such as an workplace group typically contains an open space for desks, a kitchen, seminar rooms and toilets) and proximities to either key spaces in the building like an entrance or site conditions such as orientation. Additionally, each unit contains a set of rules for changing state like location or shape (*behaviour*). The model only knows about its immediate neighbourhood and nothing about global conditions. Performance is distributed across all the acting spatial units as an agency of the designer who does not set explicit external performance targets but observes the system struggling to resolve his assumptions. It could be argued that the system of units reflects the understanding of the observer about the design issues at hand.

Two types of topological local assemblies have been explored, employing different behavioural agencies: sequential and simultaneous assemblies. Three types of computational meta-heuristics are employed whose behaviours correlate to design heuristics and project objectives: agent-based systems, nested graphs and physical force simulation.

5.2.1 Sequential Assemblies

Sequential assemblies represent the most basic and humanly intuitive approach since a designer's traditional heuristic builds on the linear composition of units into a global configuration. In a professional design process a topological approximation of a building configuration is usually developed first based on the accommodation schedule including the adjacency matrix, resulting in an adjacency diagram, colloquially also known as *bubble diagram*. Sequential models use an order starting with a specific spatial unit at a location and work through a sequence of the schedule, based on some kind of hierarchy often representing the operational or structure of the client's organization.

LINKED ELEMENTS

The simplest of cases constitutes an array of elements that are procedurally aggregated based on some simple adjacency rule such as linking corridor elements. Such a basic model was explored by CDR in 2005 for a hospital layout prototype. The prototype only defined one spatial unit as a relation between a nursing station and three patient rooms, which follow specific constraints of

a. Ratio 1 nursing station – to – 3 patient rooms
b. Distance <= 8 meters door-2-door
c. Angle <= 30° for visual supervision

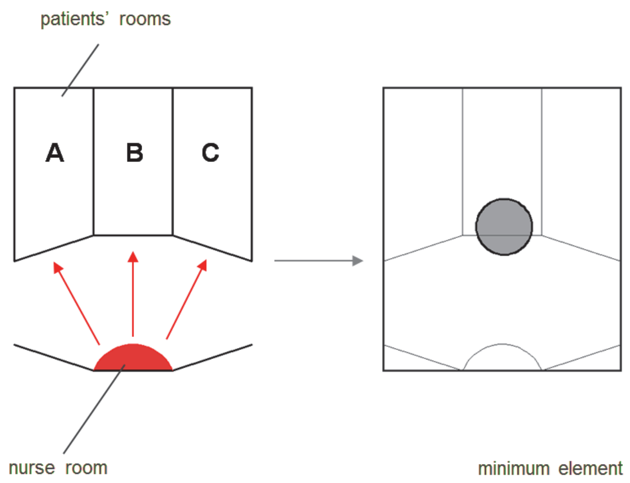


***Fig94**. The spatial unit for the hospital assembly showing configuration between nurse and patient rooms*

The centre of this aggregate unit lies on the circulation, connecting on two sides, which give the spatial unit an orientation and therefore geometric bias. An initial unit was placed arbitrarily within a site boundary and additional units added themselves sequentially until a maximum distance was reached (fire egress) or a site boundary was crossed. In either case, a placeholder was inserted with a minimum footprint to allow for a core to be inserted. The units kept arraying into linear assemblies, connecting the corridor sides until both terminal units would intersect with the assembly or the boundary. Here, the designer's constraints as regulation (ratios) and concept (linear assembly) were coded into the simple spatial unit as geometry and behaviour. Many solutions could be generated that had to be visually inspected by the designer for further elaboration.

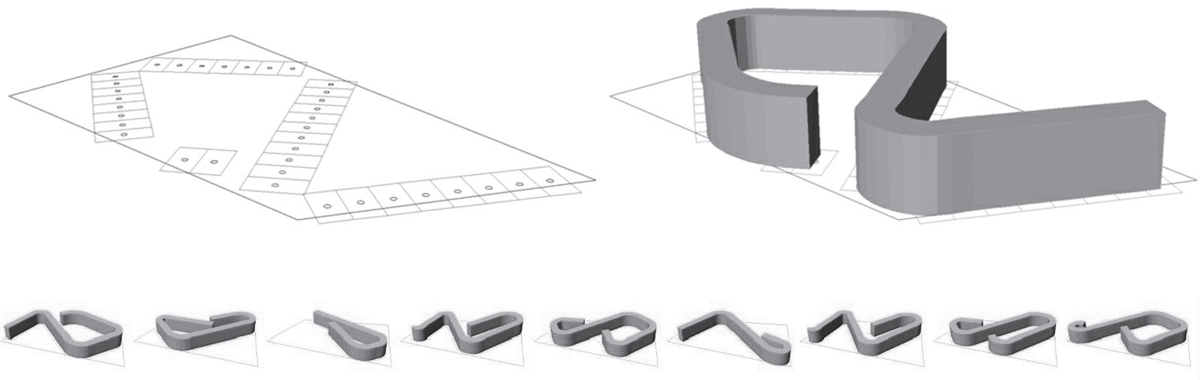


***Fig95**. The hospital assembly sequentially executed growing various linked morphologies as result that respond to the site boundary with turning angles set by staircase dimensions*

SERIAL BEHAVIOUR

From the simple hospital prototype a more complex model emerged that integrated more constraints and design heuristics. The model aimed at approximating retail unit compositions like malls that work with a simple key driver: Rentable area vs Circulation performance. The value of the overall rentable areas was to be calculated as a consequence of a generated circulation diagram. For retail, the circulation is not meant to be complicated, which means that cycles (loops) are generally avoided, creating often linear circulation trees that connect access points from perimeter to some central atria. This basic assumption was encoded into an agent-based algorithm, where each agent represented a manually seeded access point along the perimeter. Access-agents would try to move towards the other access agents by using three behavioural principles of Graig Reynolds' steering behaviours (1999):

a. Cohesion centre of gravity between all access agents = giving direction
b. Arrival distance to potential encounter with all access agents = speed
c. Pursuit modified Cohesion to pursue moving targets for Arrival

While cohesion determines the overall direction of the agents by calculating their combined central position at any time, arrival adjusts each agent's speed by a correlation between distance and movement segment. This allows agents that are closer to the estimated arrival location to move slower towards the encounter point and those further away faster, so that all meet simultaneously. Pursuit helps to adjust the direction at each step and adjust the Arrival speed as the Arrival location is moving over time as all agents change locations constantly.

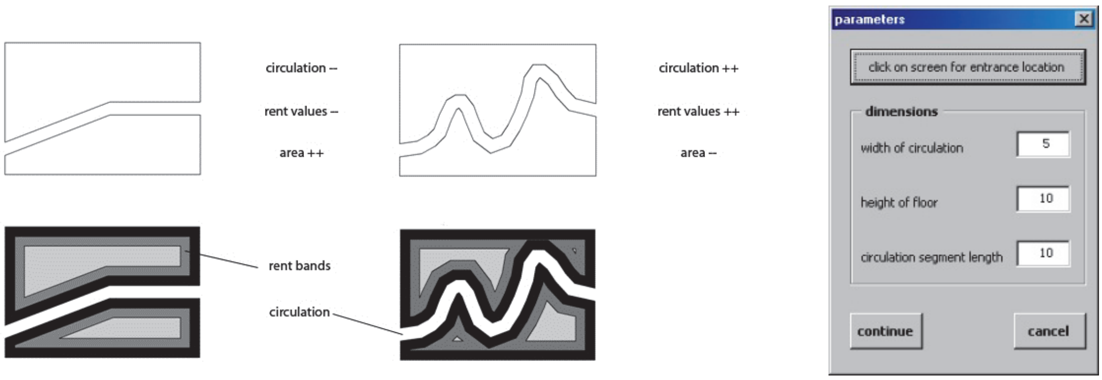


***Fig96**. Set up of the retail complex generation model: trade-off between shop frontage and rentable area (left) and the GUI for the agent-based model allowing the weighting of dimensional criteria (right)*

Two correlations to design drivers are embodied in those algorithmic behaviours: 1) Arrival and Pursuit generate a communal encounter location analogous to an atrium; and 2) the Arrival and Pursuit principles 'turn' the direction of the agents slowly 'into' the development plot, analogous with the retail design heuristic that visitors want to look 'into' a mass (Lopez, 2003). Additionally, the resulting circulation trees produce a balance between sinuosity to generate more exposed internal elevation for shop frontage and maintaining larger cohesive areas to allow for rental depth. The assumption for rental bands was a 5 meter offset per value band, decreasing from shop frontage inwards.

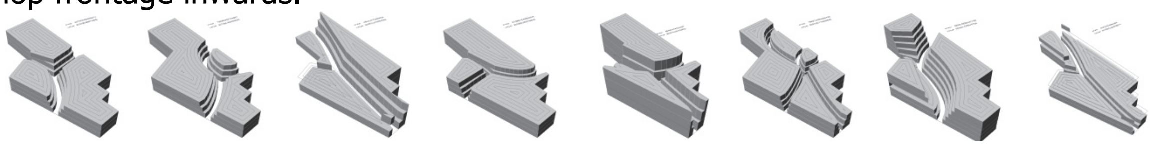


***Fig97**. Carved-out circulation-based retail massing options generated from weighted agent-behaviours and building constraints*

The steering behaviours carving out rental areas constitute an inverse planning process from industry where flows lead to layouts. Although the flows of agents appear rather literal they were not used as abstractions of 'shoppers' but as an epistemic analogy to the implicit heuristic of the retail designer who provided discursive input drivers. The resulting sequential aggregates served to select an entrance location strategy rather than a morphological diagram. As such they constitute topological diagrams of access configuration. As opposed to common aggregation, this model represented the agents as spatial units of circulation to sequentially 'dissect' a site into a configuration.

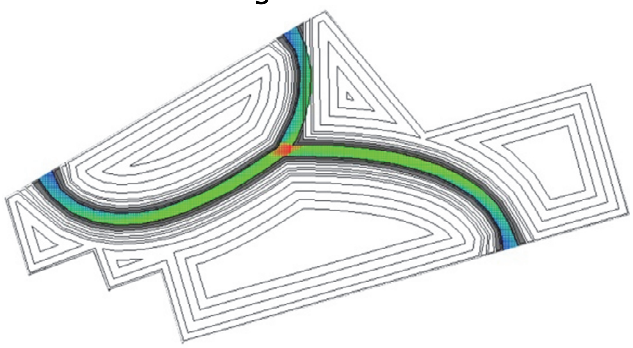


Fig98. *Resulting circulation diagrams were evaluated by isovist fields analysis to attribute value to shop fronts*

NESTED SPACES

The retail model represents a specific case of circulation that is not easily transferable to other building briefs due to its correspondence between bespoke design drivers and algorithmic behaviour. The concept of topological adjacency diagrams is however based on generic input data and generic design behaviours. In 2006, Pablo Miranda of CDR developed a sequential adjacency simulation aiming to be as general as possible, which was simply called Adjacency Diagram tool (ADT).

The ADT represents the second generation of three generic bubble diagramming tools (first and third are simultaneous assembly models and are treated in the next section). While the previous two sequential assemblies were based on typology specific heuristics, the ADT aimed not at the design behaviour of the observer but at a generic understanding of building configurations. Strategically, it was hoped that the uncoupling of behaviour from organizational logic would allow the application of any heuristic to a topological assembly. The typical bubble diagram is in fact a deliberate representational reduction independent of typology, aiming at topological proximities rather than assembly heuristic. A building is broken down into spatial types ordered in the accommodation schedule according to operational room groups and hierarchies between those groups. Adjacencies indicate group internal and building internal preferences for allocation of spaces. Manual bubble diagrams require the placement of a first room or group to assemble onto. All room types and groups are contained within a kind of super-group called the building, which is never explicitly specified. It was felt that this ontological description of a generic building could be used to develop automatic bubble diagrams where the visually unfolded diagram sits side-by-side in the GUI with the numerically tabled logic that the architect can manipulate.

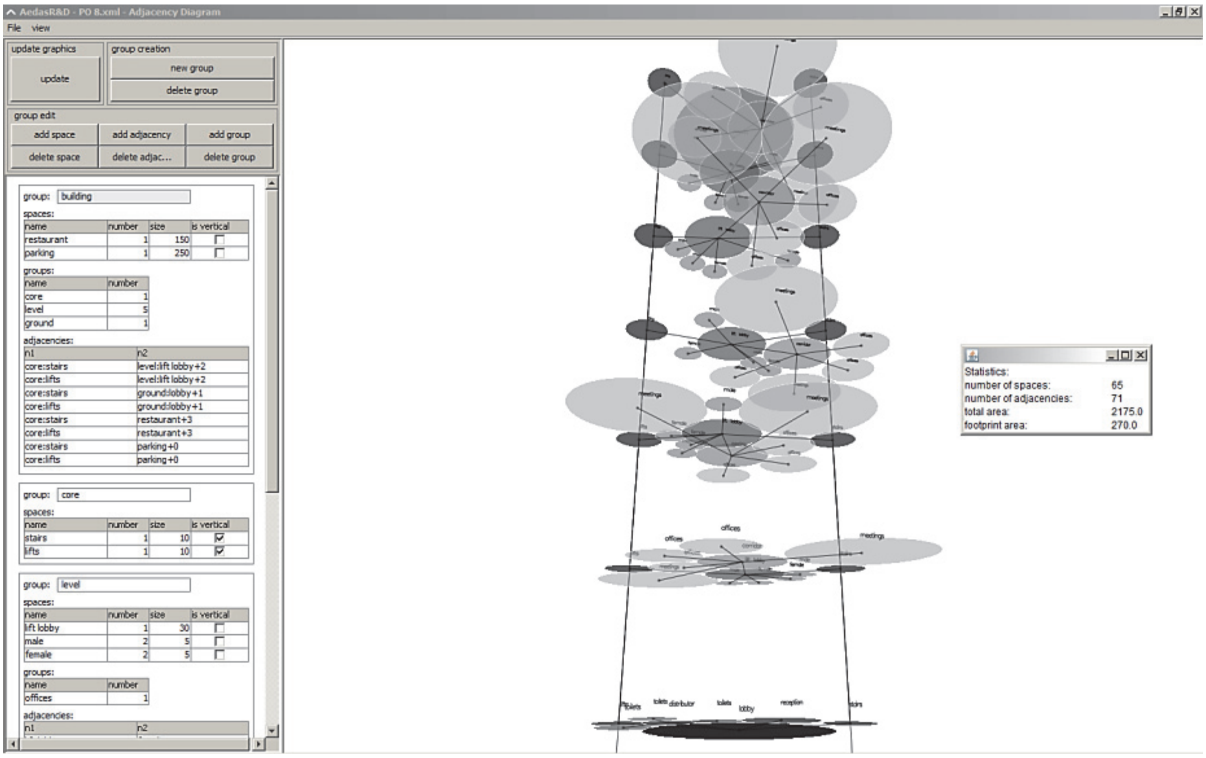
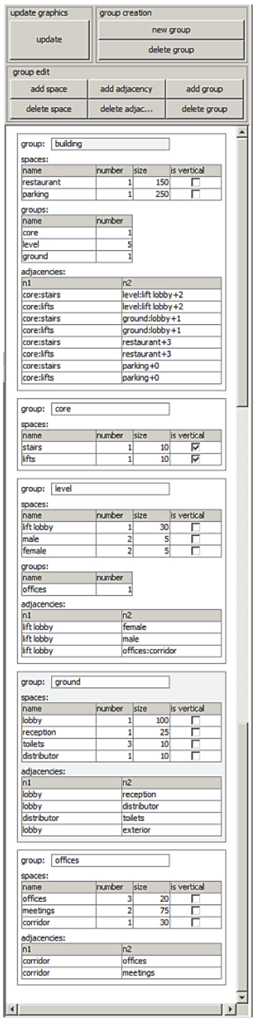


Fig99*. The Adjacency Diagram tool (ADT): left the tabled accommodation schedule as interactive interface and right, the visual representation of the recursively nested traversal graph of the schedule (grey scales indicating room depth within topology from a root node)*

As this ontology is a topological description of connections between entities, an isomorphic algorithmic model was sought that processes logically (connections) rather than spatially (locations). A graph-theoretical procedure called *graph traversal* was chosen that incrementally assembles a tree graph by recursion. To produce cycles (circulation loops as in networks), two trees could be connected via common nodes. Akin to a production system for shape or graph grammars, graph traversal iteratively inspects each node sequentially and executes an adjacency if specified in the tabled data. A root node exists from which to start traversing and two types of iteration are generally used: depth-first or breadth-first. It was decided that the depth-first algorithm aligned better to the ontological development of a building logic and the analogue procedure of a 'bubble diagram' as the main topology is first implemented before secondary rooms. This also follows a progressive scale resolution where first groups and then detailed rooms layouts within groups are resolved. The equivalent in graph traversal to main topology is called the *main graph* and the resolution within a group is called *sub-graphs*. The depth-first traversal follows the main graph first by assembling the 'children' spaces or vertical connectivity like biological generations. Then it recurses back upwards to the main graph (backtracking) to the first group to fill in the 'siblings' within each group, i.e. the horizontal connectivity within each generation. This means that the recursively generated building connectivity diagram is isomorphic to the nested representation of the tabled spatial definition.




```xml
<GroupDeclaration Name="building">
<Space Name="restaurant" Number="1" Size="150.0" Vertical="false" />
<Space Name="parking" Number="1" Size="250.0" Vertical="false" />
<Group Name="core" Number="1" />
<Group Name="level" Number="5" />
<Group Name="ground" Number="1" />
<Link n1="core:stairs" n2="level:lift lobby+2" />
<Link n1="core:lifts" n2="level:lift lobby+2" />
<Link n1="core:stairs" n2="ground:lobby+1" />
<Link n1="core:lifts" n2="ground:lobby+1" />
<Link n1="core:stairs" n2="restaurant+3" />
<Link n1="core:lifts" n2="restaurant+3" />
<Link n1="core:stairs" n2="parking+0" />
<Link n1="core:lifts" n2="parking+0" />
</GroupDeclaration>

<GroupDeclaration Name="core">
<Space Name="stairs" Number="1" Size="10.0" Vertical="true" />
<Space Name="lifts" Number="1" Size="10.0" Vertical="true" />
</GroupDeclaration>

<GroupDeclaration Name="level">
<Space Name="lift lobby" Number="1" Size="30.0" Vertical="false" />
<Space Name="male" Number="2" Size="5.0" Vertical="false" />
<Space Name="female" Number="2" Size="5.0" Vertical="false" />
<Group Name="offices" Number="1" />
<Link n1="lift lobby" n2="female" />
<Link n1="lift lobby" n2="male" />
<Link n1="lift lobby" n2="offices:corridor" />
</GroupDeclaration>

<GroupDeclaration Name="ground">
<Space Name="lobby" Number="1" Size="100.0" Vertical="false" />
<Space Name="reception" Number="1" Size="25.0" Vertical="false" />
<Space Name="toilets" Number="3" Size="10.0" Vertical="false" />
<Space Name="distributor" Number="1" Size="10.0" Vertical="false" />
<Link n1="lobby" n2="reception" />
<Link n1="lobby" n2="distributor" />
<Link n1="distributor" n2="toilets" />
<Link n1="lobby" n2="exterior" />
</GroupDeclaration>

<GroupDeclaration Name="offices">
<Space Name="offices" Number="3" Size="20.0" Vertical="false" />
<Space Name="meetings" Number="2" Size="75.0" Vertical="false" />
<Space Name="corridor" Number="1" Size="30.0" Vertical="false" />
<Link n1="corridor" n2="offices" />
<Link n1="corridor" n2="meetings" />
</GroupDeclaration>
```

***Fig100**. The two modes of inputting and editing the accommodation schedule: left the GUI nested table and right the XML definition (which can either be an input or a saved output)*

The accommodation schedule could be imported via an XML or filled in directly in the GUI via a tabling format reflecting the nesting logic of room groups. Each group contains rooms, nested groups and adjacencies. If a room from another group needs to link into a group, then it is nested inside the group and linked via room types. In the shown example, an office group corridor space links into the core group via the lift lobby space

core:lift_lobby = office:corridor

A specific allocation for a group on a floor can be given via the abstract main group, called building. This group can have individual spaces but generally comprises the container within which all groups are connected. Without the 'building group' there is no building configuration. A completed project description from the GUI could be saved and opened for editing.

The visualization of the diagram reflects the graph logic by showing a simple link connection as a line where two rooms are specified to be adjacent. Rooms are simply visualized as circles with their areas defined in the accommodation schedule. When the adjacency diagram has been generated a topological depth is calculated showing the level of nesting as a grey scale. Black shows depth = 0 and light grey = highest depth, with the 'lobby' space in group 'ground' being the root space of calculation. Topological depth is a measure introduced by Hillier and Hanson (1984) to indicate accessibility criteria and hierarchy. The shallower the depth of a space, the more accessible it is from a given root space. The deeper a space is situated within a topology, the more remote and hence possibly more private it becomes. In the example project, an intuitive contradiction occurs, because geometry is not equal to topology and as a consequence the restaurant on level six has lower depth than the meeting rooms on level one. Depth therefore, measured only topologically, might not account for true perception of depth and hierarchy, as staircases or lifts can be measured in many ways and other operational measures such as security dramatically changes the actual use.

Although the visualized adjacency diagram is a graphic expression of a temporarily fixed data set, it is possible to change the location of individual bubbles in space through simulation of physical interaction by dragging the room circles in the interface. A virtual force-directed spring system has been applied to the geometric visualization based on Kamada-Kawai's algorithm (1989) that applies a spring force to connected nodes proportional to their topological distance in the graph. The visualized lines in the diagram show the observer where attractive spring forces are applied (Helme, Derix and Izaki 2014). A repulsive force is applied to all unconnected nodes within each level in order to avoid overlaps of room circles. When the observer interacts by manually changing the location of a node, the geometric diagram will try to solve this proposal spatially without changing the tabled adjacencies. Hence, the design space becomes much larger than just the specified topology. Concurrently, the topology can be modified in the GUI tables and updated. An extension of ADT allows the observer-designer to link multiple XML accommodation schedules by embedding room groups into each other and compile them in the visual interface. That way, complex buildings can be treated as nested sub-complexes.

5.2.2 Simultaneous Assemblies

Sequential assemblies rely on a hierarchy and order in the application of spatial units. The initial condition constrains the consecutive options as described by Hillier's *barring process* (1996). But if all units were to start and iterate simultaneously, this hierarchy would be largely avoided (largely because there is no completely neutral initial condition as starting with all units simultaneously also requires them to be initially placed in space – even if that is a common zero point). When this order is omitted, the algorithmic logic changes from a continuous time-based aggregation to a pseudo-parallel[33] negotiation, where discrete time replaces continuous time (and therefore no history exists, similar to a Markov Process), and only two states exist - current and future state so that

myState(future) = function(myState(present))

That means in a sequential assembly, positioned spatial units lose their agency, while the units in simultaneous assemblies retain their agency as long as no halting

[33] 'pseudo-parallel': the computer as Turing machine and built on the van Neumann architecture is a serial computer, so that its structure cannot physically compute in parallel

condition or equilibrium has been achieved. All units are persistently trying to attain a satisfactory state in their local environment, representing a synchronous distributed computing model that Paul Coates called an *illustration of consensus* (2006, p6).

In practice, this approach has no precedence as analogue adjacency diagrams have always been solved in sequence by an individual or a group taking turns. In Robin Liggett's list of Automatic Facilities Layout models (2000), none of the models encode a distributed synchronous computing mechanism (local construction is proposed like evolutionary algorithms and pair-wise exchange but not simultaneous construction), because it appears that academic models up to that point attempted to replicate the whole design stage of layout planning with all its decisions phases – sorting hierarchies, topological diagram, geometric embodiment with areas and site constraints – instead of focussing on a core problem: solving adjacencies topologically. If the design heuristics for solving adjacencies at the bubble diagram level are taken in isolation, it becomes obvious that the designer would benefit from a simultaneous assembly as even his analogue heuristic represents an iterative adjustment of each unit to absorb the impact of a previous unit placement, albeit in sequence.

Paul Coates had been experimenting since 1991 with agent-based design models where each agent would represent literally an agency of the system that the observer tries to generate. Where in a sequential model the heuristic of the observer is designed into the global system with the units only partially autonomous, in simultaneous models every unit encodes the behavioural heuristics of the observer without a global system specification. Each unit computes autonomously containing the algorithm correlating to the observer heuristic.

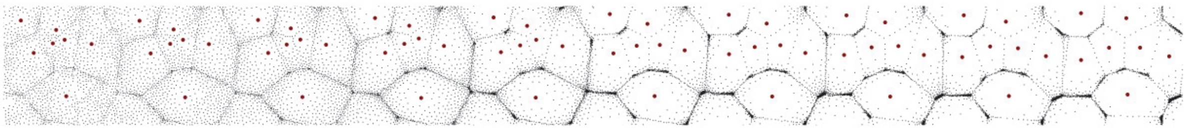


Fig101. *Coates (2010), complex geometric phenomena like the Voronoi diagram generated from multi-agent systems*

Initial agent-based experiments echoed Seymour Papert's turtle graphics (1980), which generate global form from a population of simultaneously acting agents. Coates (2010) demonstrated through a series of experiments how global geometric phenomena can be generated from local behaviours without any description of the emerging global phenomenon being provided to the agents or the system. Students of Coates and the author later showed how such distributed agency can be differentiated into classes of agents to generate sophisticated complex geometries such as blobs and precise building structures, as demonstrated in Abulmajeed Karanouh's master thesis (Coates et al. 2005).

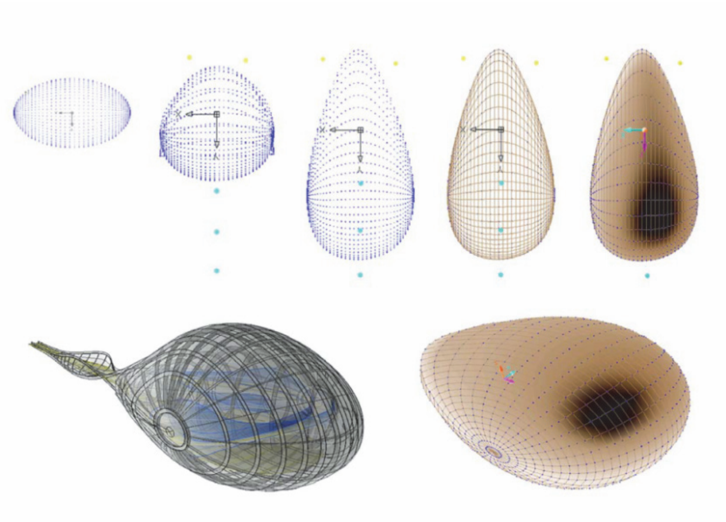


Fig102. *Controlled complex geometries using agent-based attraction-repulsion algorithm (Coates et al. 2005)*

The algorithm underlying those formations was mainly based on the *attraction-repulsion* principle[34] where each agent creates a temporary topology at each discrete time step by searching for neighbouring agents within a radius and sorting them by distance into an array. Each agent then executes its behaviour by checking a distance threshold to each of the agents in their topological list. If the neighbouring agent is too close, it will back off (repel) at the next discrete time step; if on the other hand the neighbour is further than set by the threshold, it will move towards the neighbour by a set increment at the next discrete time step. As the agent is looking at its neighbours in a frozen state, the future state is most likely a sum between the directions towards/away from the neighbouring agents:

$$X(t+1) = \sum_{n}^{neighbours=0} X(t) + step * (dis(n) * threshold\_multiplier)$$

If only one class of agents was defined with a single attraction-repulsion force, a regularly spaced mesh of agents would result. Therefore, at least two classes of agents need to be specified (or individualized behaviours) to differentiate the field into configurations (or types of rooms in an adjacency diagram). For the most simple application like the CECA teaching algorithms, a distinction was made between spatially fixed agents as attractors without behavioural agencies and mobile agents that repel each other but are attracted to the attractors. In three dimensions this generates various forms of blob, subject to weighting of attraction-repulsion forces.

[34] At CECA we named the algorithm 'attract-repel'. Officially, the formal definition is called *force-directed graphs* or layouts, where the attraction force is attributed to a connection edge and the repulsion force attributed to geometric objects like vertices (Eades 1984).

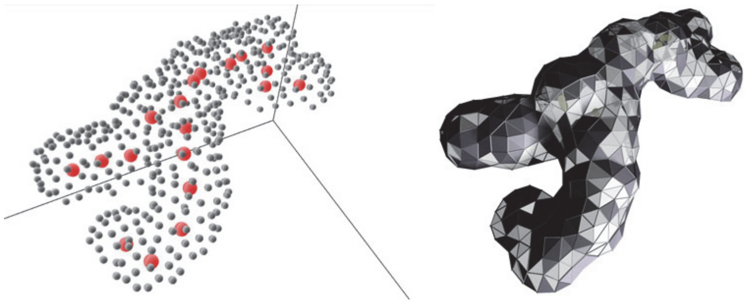


***Fig103**. Blob-like spaces using agent-based attraction-repulsion algorithm (Coates 2010)*

For adjacency diagrams in a professional context, agents do not represent a geometric element of a shape but a spatial position with a normalized area. For the first attempt to apply the CECA research to practice in 2005 at Aedas, an equivalent number of agent classes to room types had to be specified. Test cases proposed single floor educational buildings or campuses, with or without site boundaries. A series of graphical user interfaces (GUI) had to be provided, because the bubble diagram tool was plugged into AutoCAD as a VB macro, so that designers could select the model from a drop-down menu in the AutoCAD application window. The room adjacency matrix could be compiled via a GUI or imported Excel table, where connections between rooms were specified by 1 = adjacent, 0 or 'none' for neutral and -1 for remote (undesired proximity), which gave the agent classes their behavioural rules. As opposed to the CECA attraction-repulsion models, asymmetrical connections could be specified via the adjacency matrix that overcomes some of the limitations of solving adjacencies manually. In a manual process, connections are solved in series and contradictory relations would result in an infinite loop unless one waives the problem. Through parallel computing, a location can be found that resolves contradictory connections by simultaneous evaluation. An additional buffer distance was introduced to set an overlap between room bubbles, geometrically embodied via simple circles, whose radius was given by the room area.

While the model found some applications, there were many obvious practical problems that result from the algorithm, especially local minima that occur mainly due to initial seeding distributions or circular connectivity. Three possible solutions to these problems might include a sequential model like the above discussed sequentially nested (graph) model, which was in fact developed as a response to the here described attraction-repulsion model (and therefore, chronologically succeeds here); complicated mathematical heuristics like quadratic assignment functions or complex additional multi-criteria optimization as described in section 5.1 or more interaction to allow the user to interfere with local minima. As already noted above, in an intuitive *designerly* context, neither the sequential nor the mathematical or optimization model are feasible for lack of observer integration, visual clarity and heuristic correlation. A combination of parallel computing with interactive support appeared the most coherent strategy.

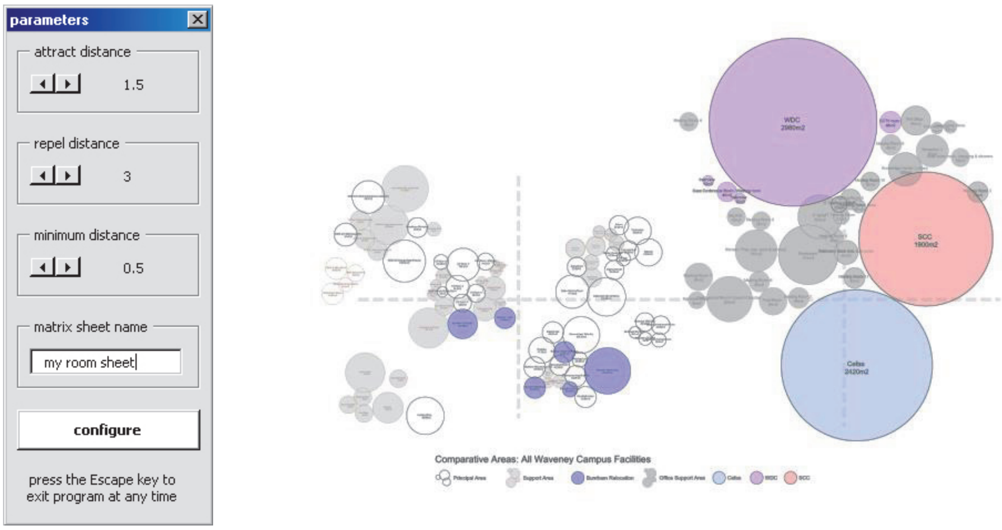


***Fig104**. Translation of the attraction-repulsion algorithm for adjacency diagram generation at Aedas 2005 with a GUI for general use by architects*

FROM BUBBLE DIAGRAMS TO MASSING DIAGRAMS

In 2009, CDR participated in a building design competition for the Abu Dhabi Education Council with Lucy Helme developing the model under the author's guidance. The competition served as a test bed to build a hybrid design process for conventional and computational design procedures. The architects compiled an accommodation schedule with adjacency matrix and provided diagrams for a spatial as well as climatic control strategy. An intended formal representation was agreed to be based on an orthogonal composition, correlating to an application of the attraction-repulsion model with a simple orthogonal geometric embodiment. CDR developed the design configuration 'engine' while the architect created input numerical values and architectural plans. Input was read from a CSV file and the output consisted of dxf (drawing interchange format) file formats of wireframes of stacked massing blocks with attributed room or area types, which had to be elaborated by the architects into detailed building plans.

The general principles of the previously discussed attraction-repulsion algorithm were maintained. Unlike the Kamada-Kawai algorithm (1989) of the ADT, forces acting along connections are not proportional to their topological but geometric distance. Attraction forces were applied to desired room adjacencies and repulsion forces to the vertices of all rooms, so that no overlap would occur. Instead of using a constant force as in the previous model, a scalar force was used called *Hooke's Law* that is proportionate to the length of the connection (or dragged location when interaction is applied – see Fig105), allowing for adaptively scaled responses to avoid either too much or too little adjustments (Arvin and House 2002).

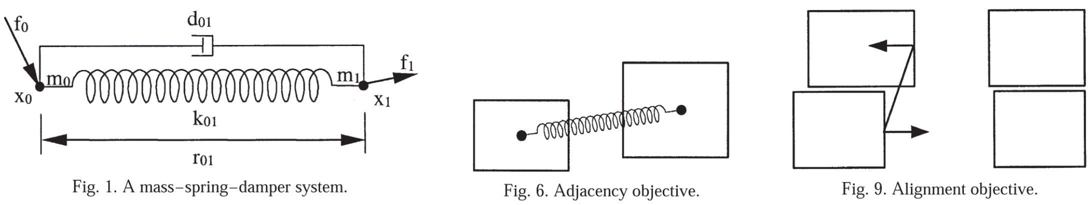


***Fig105**. Elements of the physically based space planning model of Arvin and House (Arvin and House 2002). Similar concepts are used in our model, although the implementation is different.*

Some general features were added that initially seemed project specific but later turned into generic functionality: each room of the accommodation schedule was geometrically represented by a rectangle polygon, whose proportion was given. In a competition setting time is a key consideration and the translation from a purely topological bubble diagram into a geometric interpretation would have taken too long. The designer is given an immediate approximation of the areas occupied on each floor. This required a break-down of each unit of the accommodation schedule into small abstract units of area instead of defining conglomerate spatial entities. For example, an open office area was broken down into many workspace-like cubicles rather than an open floor definition. This allowed aggregations of many small units into non-convex area definitions.

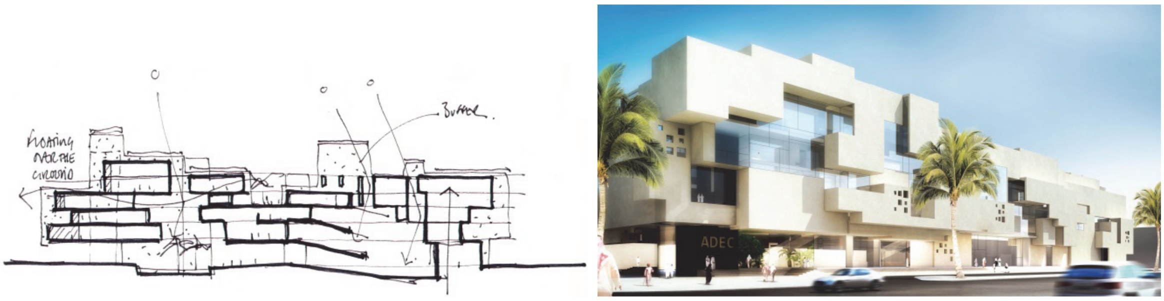


***Fig106**. Architect's input as programme sketch (left) and rendering of the CDR model (right) (Helme et al. 2014)*

Eventually, some project specific additions include an offset of the outer perimeter of each floor aggregation to approximate a second skin as climatic buffer. 20% extra area was added to all spaces providing for redundancy on each floor to enable the designers to manually extract circulation spaces. Circulation has always been a problem for 'layout automation' and neither Liggett's survey (Liggett, 2000) nor Arvin and House's precedence (Arvin and House, 2002) have attempted to solve them. Ulrich Flemming's LOOS (Flemming et al 1992) attempted to approximate abstractions of circulation and unassigned spaces by allowing for *loose* packing and identifying *non-trivial* holes. But the abstraction rather represented left-over space and offsets between adjacent rooms than meaningful circulation diagrams. Even more recent academic research such as Tomor Elezkurtaj's interactive evolutionary approach simply ignores circulation (Elezkurtaj and Frank 2002). CDR models with circulation are discussed below. Eventually, the model approximated an overall gross-floor-area (GFA) that when compared to the prescribed maximum envelope extents of the competition brief highlights the 'space-left-over-after-planning' (SLOAP). SLOAP is generally perceived as a negative measure for redundancy in urban planning but can also be regarded as an overseen development opportunity for additional use of external space or higher density of the area schedule.

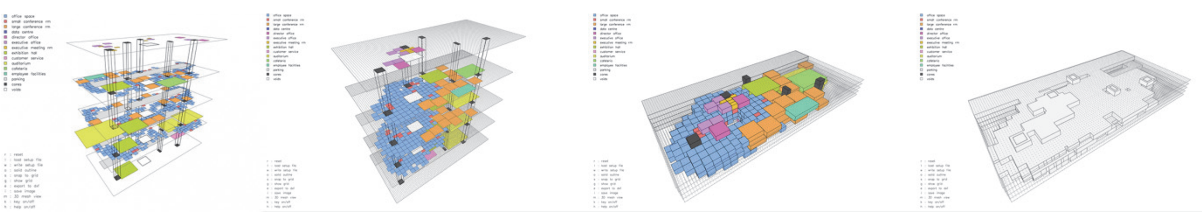


***Fig107**. Semi-automatic massing model for ADEC showing from left to right: initial random state of rooms on floors, an ordered model after organization, the block model and adding the double skin (Helme et al. 2014)*

The model took the logic of the previous force-directed layout concept and added the interaction from the graph-traversal model. A designer can visually follow the dynamic position changes of the room polygons as they are searching to fulfil the adjacency requirements. As no constraints were given for orientation but only proximity to fixed cores and mutual adjacencies, aggregations fall into equilibria that are always local minima. The interaction allows the designer to select a room square any time and drag it to a desired location. As the process is dynamic, all adjacencies are fluidly updated in real-time, immediately illustrating the consequence of this interference as all rooms absorb the interaction across all floors. The model therefore represents a heuristic synthesis between algorithmic and analogue drivers where each actor – computational model and designer – observe and absorb each other's changes of state and intentions. This is done without a formal structure, in the sense, that no translation of this adaptation is provided via some explicit mapping mechanism of one actor's action onto the other. A designer-observer only acts on the positions of rooms interactively, not on the behaviour or ontology of the units themselves.

GENERALIZING ADJACENCY DIAGRAMS

The ADEC model received positive feedback from architects inside and outside of Aedas as the correlation between design and algorithmic heuristics were intuitively apparent[35]. While the ADEC model constitutes a step within the evolution of CDR's semi-automatic adjacency diagrams, it represents a prototype whose algorithmic system and peripheral components like user-interaction and user-interface only needed refining while additional functionality could be added to generalize it for generic use as a layout tool. Feedback was gathered through a series of presentations and workshops with design teams from whom two pieces of knowledge were extracted: a) what is their heuristic for creating adjacency diagrams and massing models; and b) what changes do they propose to the prototypes? In a professional setting, the answers to those questions were logged but simply compiled for internal use and hence no record as such exists, other than the resulting developments.

The generalized model was to integrate everyday aspects of design heuristics for adjacency diagrams with an abstracted geometry for massing (hence, the term 'layout tool' as a colloquial description, not correlating to professional layout design phases). The main issue for an architectural designer accustomed to traditional digital procedures is control and evaluation. Automated processes often allow for little control and insertion of personal preference. A balance needs to be struck between automated simulation and manual process, making software sufficiently open to interaction with the automated heuristic and constraints, without

[35] The author had Tomor Elezkurtaj present his interactive evolutionary layout model in 2005 in the Aedas London office to a large group of architects to promote this type of development but interest was very low. This feedback was integrated into the development plan of CDR by the author as it was clear that Elezkurtaj's model did not correlate with the designer's heuristic: the interaction works on the geometry level of the spatial units, not on their topological positions (Elezkurtaj 2002). Therefore, the visualized re-calculation of the model shows the adjustment of geometry rather than the topological logic of an adjacency diagram.

compromising the algorithmic behaviour. Control is also exerted via evaluation to check whether a novel process complies with known objectives. The task therefore was to open up the ADEC prototype to allow for multiple modes of control and visualize more evaluation criteria.

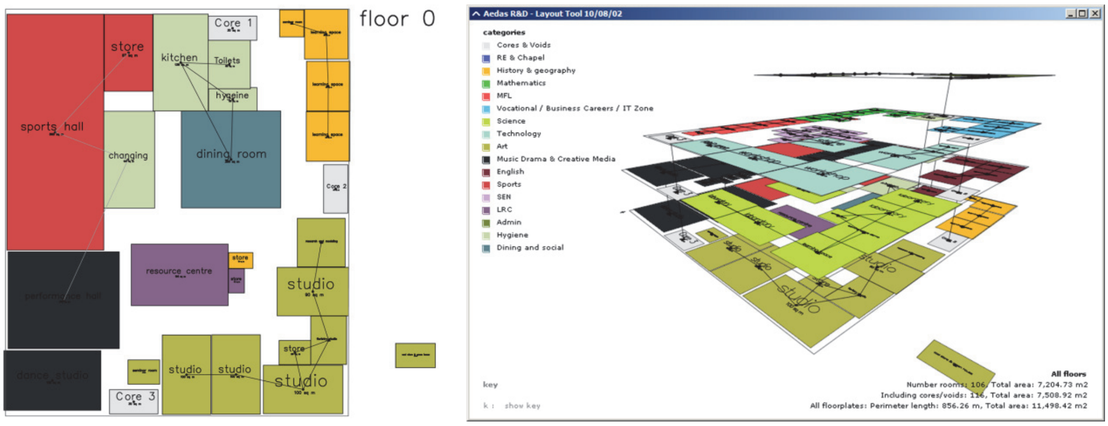


Fig108. *Physically based layout tool, applied to a school. Left: view of one floor; right: screenshot of tool in use showing floor layouts expanded in height, and readout of areas achieved.*

To increase control, perceptual aids were implemented. Topological connections between rooms with desired adjacency were visualized by simple lines, showing where attraction forces are applied. More than control, it helps the designer understand the dynamic process by revealing the internal process. Adjacency links could now be edited directly in the interface by selecting two room elements to join or disconnect, updating all room connections dynamically. The adjacency matrix can therefore be specified via an input CSV file or connected directly with visual feedback. All rooms could either be imported or seeded interactively. At any time, the current state can be exported and re-imported later for manipulation. Area specifications can be set in the accommodation schedule input file or via a dialogue box in the interface. Variables like geometric dimensions (area, proportion), area type and hierarchy (groups) with layers, dynamic properties of fixed location with behaviour, fixed location without behaviour or mobile, and floor number could be specified. Additionally, modest transformation functions were included like area splitting into sub-rooms, changing room aspect ratios and rotating area squares. Some aspects were already built into previous models, such as restricting the units' positioning to within site or floor plate boundaries. These functionalities give the designer the option to generate or explore a layout in real-time, and save to a matrix as an accommodation schedule rather than starting with numerical dimensions.

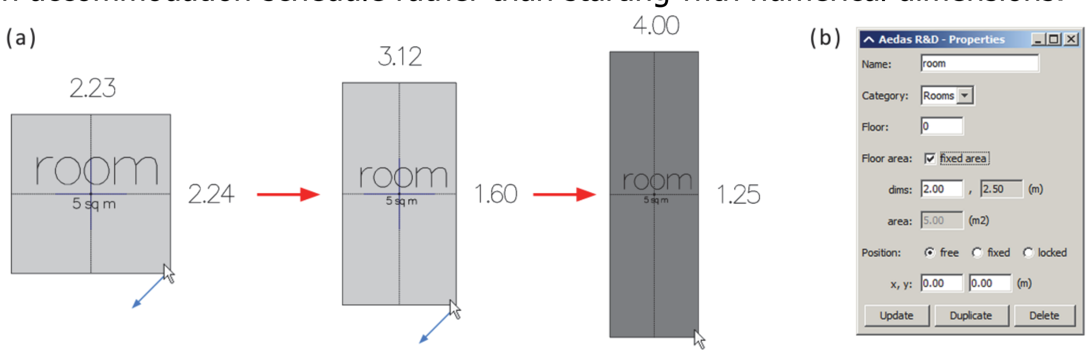


Fig109. *Direct manipulation of dimensions: (a) room aspect transformation by dragging corner of spaces; (b) changing parameters in dialogue window (Helme et al 2014)*

As this type of semi-automatic bubble diagram with geometric block model synthesis is new, the designer is given the option to switch between the traditionally separated diagrams: topological adjacency and geometric aggregation. This allows him to manually interfere with the bubble diagram and check geometric effects immediately and vice versa. Read-outs of area performances show overall net internal area (NFA), GFA and unit numbers by types.

5.3 PARTS ADAPTION | GEOMETRIC SEARCH

Section 5.1 dealt with models with explicit targets for optimization. Targets were set by the observer through explicit numerical values. Those values were a measure of a composition whose spatial units did not reflect any properties of the global performance. The algorithmic models that produce the measurable outputs are independent from the heuristic of the observer-designer and constitute a black box.

Models in section 5.2 concentrate on the units with performances embedded as an agency of the observer. These models execute the specifications of the locally acting units where global performances are derived from the consensus of the parts. Agencies represented key performance indicators that were taken from external sources such as design guidance and the designer's heuristic to inform the behavioural specification of the spatial units. The algorithmic model is isomorphic to the designer's heuristic or ontological perception of the design typology (i.e. the make-up of a spatial description as provided by an accepted typology).

Either the observer's intentions or the model's logic dominate design states. A hybrid model to generate spatial patterns is represented by cellular (or discrete) models, such as cellular automata (CA). At first sight, CAs are fully topological and traditionally cells are represented via integer states such as discrete land-uses, and often only binary states. CAs appear valuable when trying to approximate generic global patterns and from experience in practice, specific design constraints are difficult to implement. Hence, CAs are mostly applied to urban scales where development theories of large-scale land-use patterns are simulated (Batty 1994). Also the author has experimented on live projects with CAs on a strategic level for land-use distribution. But at this scale and resolution, the observer's design intentions are difficult if not impossible to mediate.

If CAs are to be applied at a lower spatial scale, the cell structure needs to change from representing integer states to the representation of the geometric states of each cell. Geometric definitions of cells require the cells not only to communicate topologically but also as a continuous space where each cell's definition has a geometric effect in space on its neighbours. A dual and mutual rule-set is now required that associates the topological structure of the algorithmic representation with the spatial logic of the observer's intentions. In the CA-specific case, state transition rules govern the adaptation of a cell or spatial unit to the condition of its topological neighbourhood. These rules encode also the design intentions of the observer. The cells have a separate mechanism for querying their topological neighbourhood that is aligned with the design purpose but is autonomous from design constraints. The observer on the other hand specifies the geometry of the cell and imposes his design purpose, constrained by the processing topology, therefore structurally determined as opposed to distinct from the underlying algorithmic logic as in the previous topological models of section 5.2. Hence, a three-way structure emerges where the state transition rules represent an associative mapping between the algorithmic topology and the design purpose as Hillier's *manifold* (Hillier and Leaman 1974) suggested. The resulting morphology represents a process field that synthesizes all three layers of the model: topology, geometry and association.

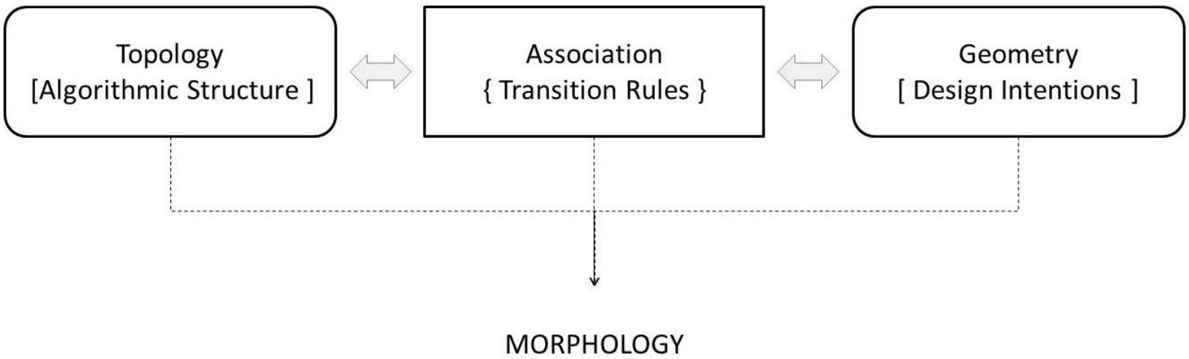


Fig110*. Structure of the cellular automaton where the transition rules map the algorithm states into design intentions and vice versa to generate consensus on morphological configurations*

5.3.1 Geometric Topologies | Cellular Partitioning

With students of CECA the author trialled a series of CAs with geometric cell definitions to generate proof-of-concept diagrammatic building floor layouts. The projects represent teaching exercises based on the author's specifications and students had to submit a project report. During the academic year 2005-6, Edward Finucane produced the first cell partition CA with the cells representing 'walls' to form corridors and enclosures in plan. The test was to evaluate emerging partitioning types without being specific about the building typology, akin to Bill Hillier's *barring process* (see 3.7.2) but in this case automating the barring or placement of partitions. The partitioning of a plane through wall sections intended to create a mesh of continuous and encloses spaces with varying level of enclosure.

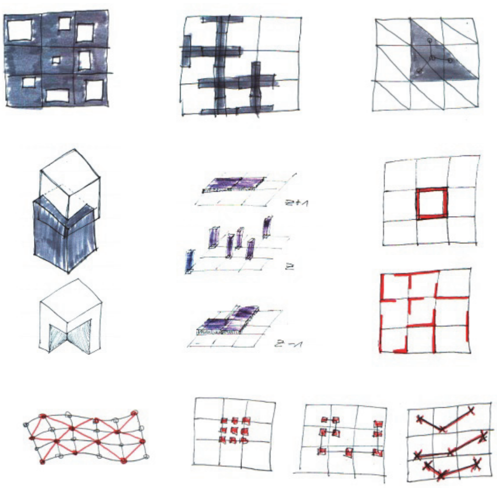


Fig111*. Typologies of cell states on an orthogonal grid sketched by author for students in 2005 at CECA*

The CA works in two dimensions and uses a Moore neighbourhood of eight topologically adjacent cells. Finucane used seven cell states that represent types of partition, where the first four states are rotations of a single type of corner partition and states 4 and 5 are rotations of a linear wall partition. This leaves only three types of partitions, whose rotations are determined by the neighbourhood survey. The state transition function was coded as an exhaustive state condition list, meaning that for all seven possible current states of a cell, all eligible neighbourhood conditions were specified for their future states.

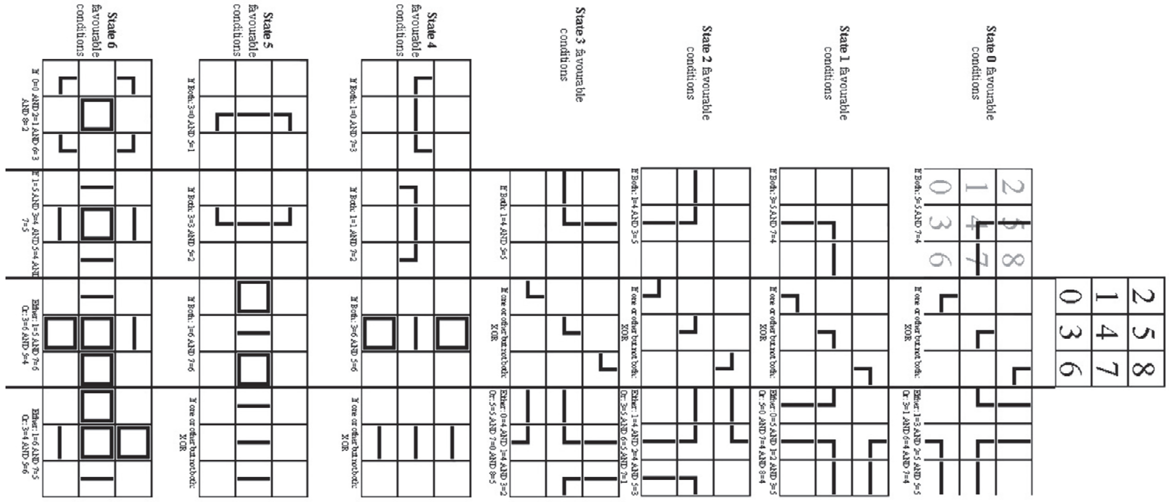


***Fig112**. The wall partition CA by Edward Finucane at CECA 2005, showing the seven cell states from neighbourhood relations*

Apart from the anticipated corridor-like and fully enclosed sub-spaces, the resulting configurations also produced semi-enclosed sub-spaces that were not initially intended. Those semi-open spaces provide a positive third spatial definition for hybrid use. Interestingly, Finucane did not provide for an empty cell but a fully enclosed cell state, which proved to be unproductive as it cannot assemble into more complex spaces and isolates the cell.

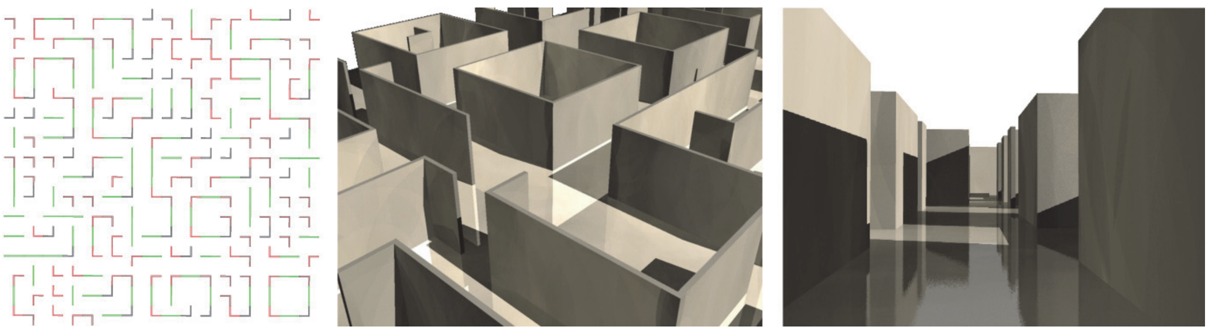


***Fig113**. Wall partition CA by Edward Finucane at CECA 2005, showing an equilibrium state (left) and two renderings*

In the following academic year of 2006-7, Phil Langley also experimented with wall partitions as cell states of a CA. As a consequence of Finucane's CA, the fully enclosed cell state was avoided and an empty cell state introduced. Only four cell types exist that reflect increasing enclosure, from zero to three wall sections. Each type can be rotated into 4 locations, creating a total of 13 states per cell (the empty cell only has 1 rotation).The sections are inserted between mid-points along the geometric connections of a von Neumann topology, which in plan only produces four connections. The mid-points of connections depend on the positions of the cell centre-points within the grid, which are also calculated from the neighbourhood survey. But for the neighbourhood survey also the top and bottom cells were queried – a 2.5 dimensional neighbourhood with 6 topological connections, so that geometrical continuity of wall alignments could be approximated in the state transition rules. Apart from the 13 states for sections and rotations, a state transition rule governs the centre-point positioning calculated from the neighbourhood. The cells therefore do not only represent geometric states themselves but also instrumentalize the geometry of the topological structure. For the spatial evaluation, the mid-point distances of the connections become essential for the definition of uses, because these define the wall lengths and therefore room properties.

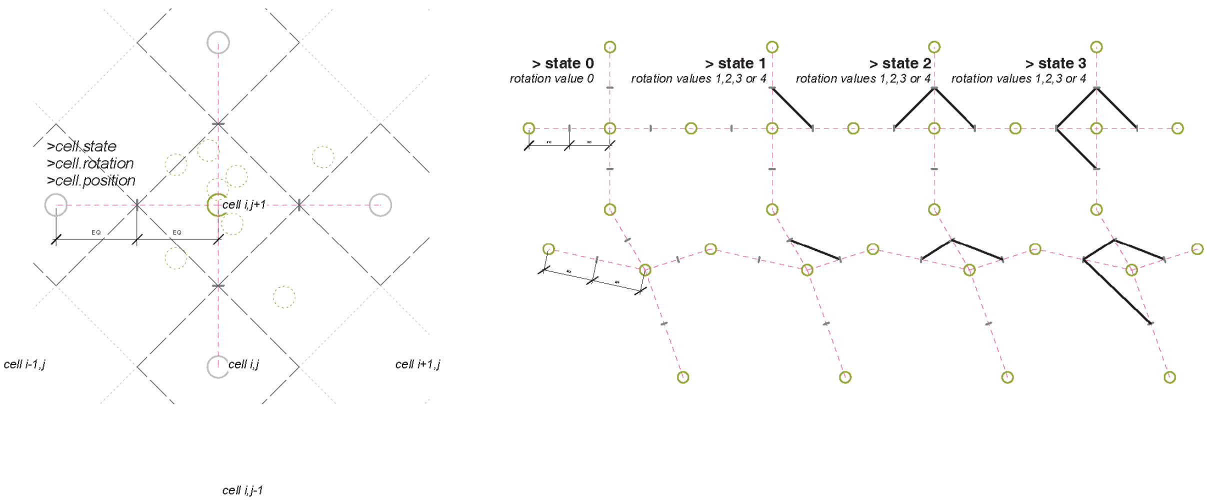


Fig114*. Wall partition CA by Phil Langley at CECA 2006, showing the grid structure (left) and four states (right)*

The state transition function was weighted so that neighbourhood counts with higher section numbers, i.e. more enclosed spaces, produce more highly enclosed spaces. The seeding of a ring of static perimeter cells with cell state type 0 (= empty), means that perimeter cells were more likely to become less enclosed. The vertical alignment rule contributed to similar enclosure types across floors. This resulted in more enclosure at the core of the configurations and more openness of spaces around the perimeter, similar to office buildings.

The center-point positioning across generations could also be constrained between no offset to a maximum offset. The maximum offset ensured that the rooms would not become unfeasible enclosures.

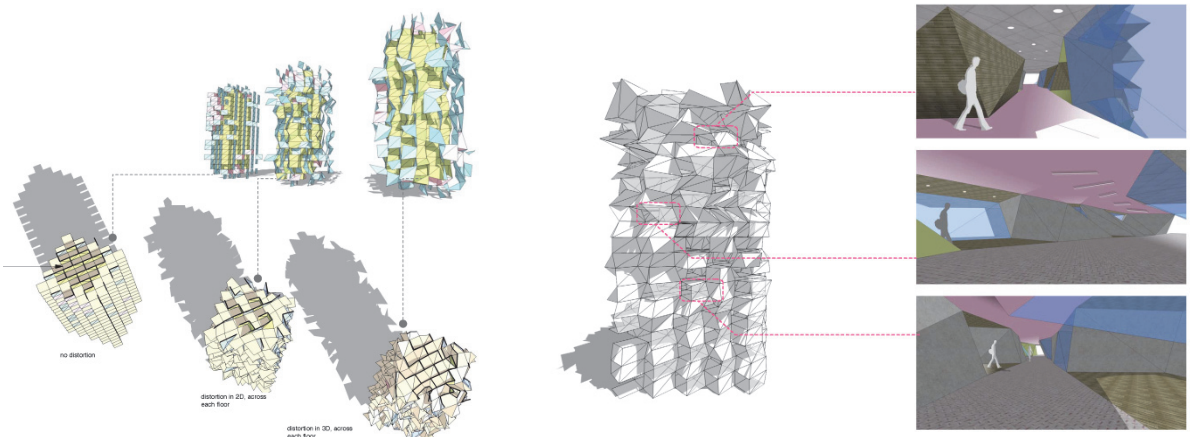


Fig115. *Wall partition CA by Phil Langley at CECA 2006, showing three generated equilibria (left) with different weightings of transition rules and (right) interior renderings*

Both CA models show how the topological structure and neighbourhood survey methods of the algorithmic logic blend with geometric definitions that approximate the observer's design intentions. In particular, Langley's model shows how the topology can be used to inform the geometry based on the same structure. In both models, the state transition function is the vehicle to mediate the observer's intentions to produce spatial layout typologies: the algorithmic model requires the observer to associate his spatial expectations to configurational rules. In Langley's model a hint of circulation is built in, which will be explored in section 5.3 where a further exploration is discussed. Both models are using a somewhat arbitrary halting function. Finucane ran the CA for a limited amount of generations that from trials approximated a dynamic equilibrium, while Langley's halting function reflected the number of floors of an imaginary building.

Corridors and enclosure types not defined in cell states. Only through simultaneous communication in field are those emerging. The state transition function allows observer to decode the underlying logic of a corridor and an enclosure-like configuration by understanding the rules for both.

5.3.2 Recursive Mediation | Hybrid Field

Like the Simultaneous Assemblies in section 5.2 CAs are synchronously processing patterns. When using geometric definitions of cells and aiming to solve a target state that does not represent a dynamic equilibrium as in a CA, orthodox consensus-driven fields are difficult to employ, mainly because a) initial seeding conditions determine global states, which then b) override any details that are not encoded in the transition function. It would be counterproductive to develop a state transition function for a CA with as many exception rules as there are meant to be design details, as this would hollow out the purpose of the passive consensual nature of a field pattern generator. Specific local design situations require cells to become active in their transition rules while still being subject to the contextual field. This hints at a hybrid process model, combining some of the behavioural logic of the assemblies of section 5.2 with the topological field adaptations of this section.

In 2007 the author was commissioned to design a modular furniture, based on the CECA teaching experiments with CAs. The client was MDFItalia[36] and the collaborating designer Massimo Mariani[37]. The furniture was specified as a modular shelving system that could be designed by clients from support by an intelligent configuration software. The initial inspiration was John Conway's Game of Life (see 3.3), providing the name VITA, and the design software's algorithm was intended to be based on a CA. The work was divided into Mariani designing the modules (or cell states) and the author with Asmund Izaki developing the design configuration system. A fixed grid size was agreed subject to standard industrial production and assembly dimensions and by recommendation of the client a structural plate was envisaged as base module for stability and ease of assembly. Hence, the notion of a topological field constraining a geometrical configuration proposed itself naturally. The software was intended for clients who wanted to design a bespoke version of a MDFItalia product, which hinted at an online configurator.

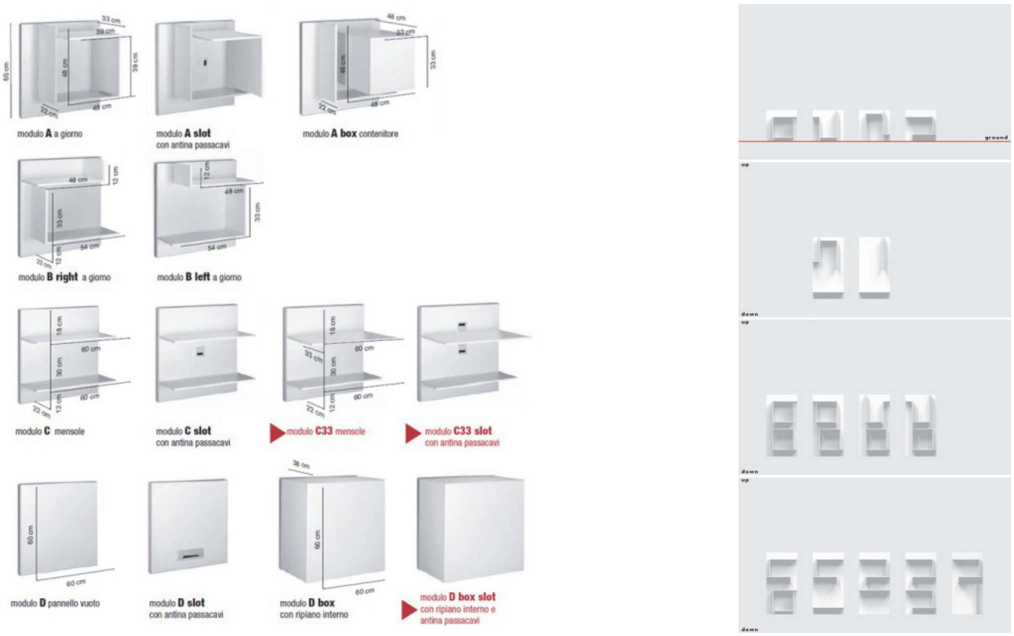


***Fig116**. VITA modules (left) and design rules for algorithmic topologies: 1) no open ends at bottom of grid to close the shelving visually and functionally towards the floor; 2) reduce open ends towards top; 3) least possible vertical shelf lines to minimize wasted storage; 4) avoid module C in vertical rotations*

The basic design concept for the shelving system was represented a continuous shelf line, akin to Finucane's partition wall CA. But instead of always connecting just two open wall segments across a cell, it was envisaged that also three open shelf segments could also be connected by a cell. The developed modules amounted to four basic states and transformations of those basic states via mirroring and rotation. Module A and C could be rotated into four orthogonal states, module B could be mirrored and rotated into four orthogonal states while module D represented an empty cell like Langley's open space cell. A total of 19 states existed per cell.

---

[36] http://www.mdfitalia.it, accessed 19.06.2015
[37] www.massimomariani.co.uk, accessed 19.06.2015

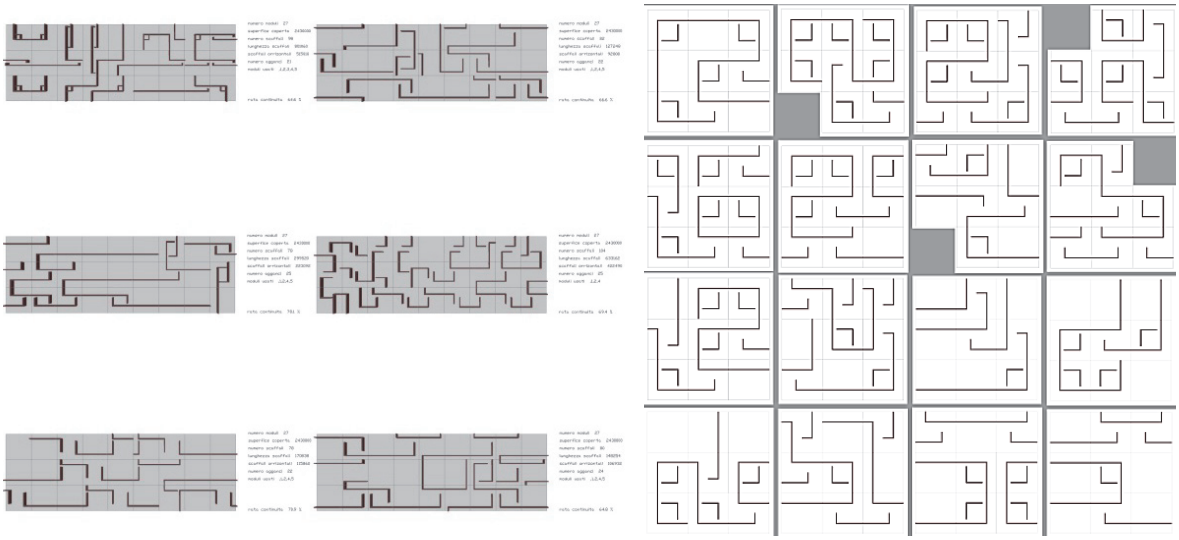


Fig117. *VITA – the first iteration programmed in VBA to trial the shelving configuration per module (left) to generate different aesthetics and the CA version on a 3X3 grid (right)*

Initial tests with a CA algorithm were promising when only aiming at general patterns to be generated from the modules. The state transition function was encoded in the most abstract possible way to account for all possible alignments within a cell's van Neumann neighbourhood: the shelf end points of each module were encoded as direction vectors and rotations. The neighbourhood survey therefore consisted of checking whether two opposing vectors (between pairwise modules) would meet at an interface point between two modules. The state of a cell consisted thus of the ratio between available connections to number of served. Despite there being 171 combinatorial possibilities on each side of a cell (19 x 19) and 684 for all sides (171 x 4), the state transition function was thus reduced to effectively five lines of code (four sides of vector matching function). The halting function was triggered by the condition of maximum internally closed end points and minimum open external end points.

Additionally, the direction vector encoding allowed swift adaptation of shelving exit points from their base plates, providing for quick testing of different offsets and their aesthetic and performance effects. But when contextual conditions, market drivers and client usability were approached, the CA was not able to handle the necessary constraints without sacrificing its core concepts. As mentioned above, the state transition function would have to be extended from its generic specification to include all possible exceptions that describe the design and market requirements. Also the nature of a CA's *voting rule*, which gives the cells their passive adaptation behaviour, unsettled stable neighbourhoods with high connectivity, so that unresolved grid cells could be resolved. The global consensus becomes an issue that points at solutions that differentiate the process field into various rates of adaptation.

An intermediate model was tested based on a variation of simulated annealing where regions in the grid could settle at different temperature rates and stabilize independently. However, simulated annealing still works on a population basis with the whole grid aiming at a consensus. The first model based on a CA was therefore used as a template for the optimization of linear connectivity.

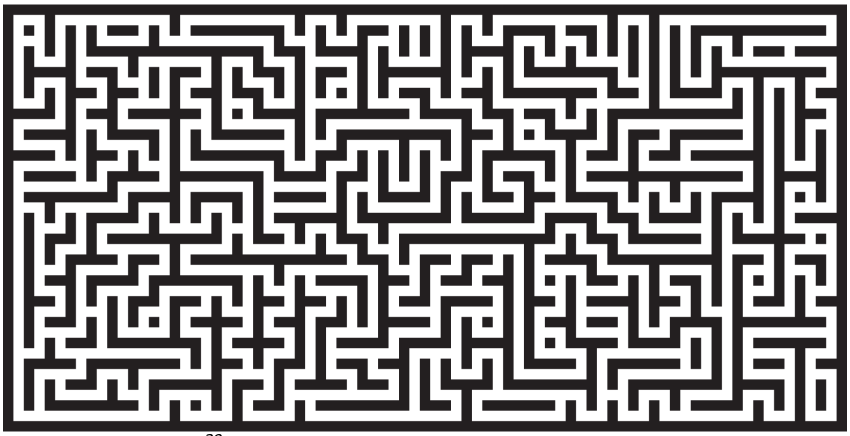


Fig118. *Computer generated maze*[38]

The concept design prototype was developed in VBA and the final version was written in Java, using Java Webstart for the online application (whereby the Java application is downloaded by a client computer from the manufacturer's server). In order to better control design and market constraints, the notion of a CA or other such parallel computing/ consensus models was abandoned in favour of a graph-based growth algorithm akin to a maze generation (Kodicek, 2005, pp. 278)[39]. Most maze generation is based on depth-first search, where a single branch is developed across the topological depth of a grid until no further growth is possible. This takes the assumption that the grid is large and the cells contain a single partition, equivalent to a single shelf per cell. But for VITA it was assumed that clients will choose grids within maximum 2-4 rows and columns, i.e. not a very deep grid; and a cell based on the designed modules will contain more than one partition thread. Hence, a breadth-first search algorithm was developed that prioritizes neighbourhood completion over depth completion.

A seed module is randomly place in the grid with a random rotation, subject to design constraints, and from each open shelf ending a neighbouring cell is filled with a module in a rotation that creates a continuous connection between the grid neighbours via their shelves. From each of the inserted neighbours the procedure is repeated, so that a connected area is growing from the seed outwards. Where connections are difficult to establish, a series of modules under different rotations are tested until a state is found that connects a neighbourhood. If that fails (called *empty queue*) or a perimeter has been reached, no module is inserted leaving the topological neighbour empty. The branch along which the queue is empty is recursively backtracked to find the next open shelf end from which to continue. For speed, a branching memory is built in that prohibits the algorithm to backtrack and visit the same branch twice. If no new modules can be inserted onto the end of the grown branches, which fully connect with their neighbourhood or comply with design constraints, then a random empty grid cell is chosen to grow a new connected sub-graph.

---

[38] https://commons.wikimedia.org/wiki/File:MAZE_40x20_DFS_no_deadends.png: accessed 09.06.2014

[39] Mazes like the VITA shelving system consist of multiple connected or disconnected sub-graphs unlike labyrinths, which consist of a single graph.

Fig119. 24 version of a 3X3 grid generated by the engine and one assembled 3X3 version at the MDFItalia showroom

A shelving solution can either be generated automatically by the algorithm or the client can manually compose a configuration. Interaction allows the client-user to influence the configuration at each step, for example by seeding modules in the grid manually with specific rotations (like placing empty D modules for TV placement) that the algorithm has to integrate into a global solution; or deleting some modules of a completed configuration and allowing the algorithm to complete those sub-areas. Further interactive functionalities were added to enable the client to participate in the algorithmic system.

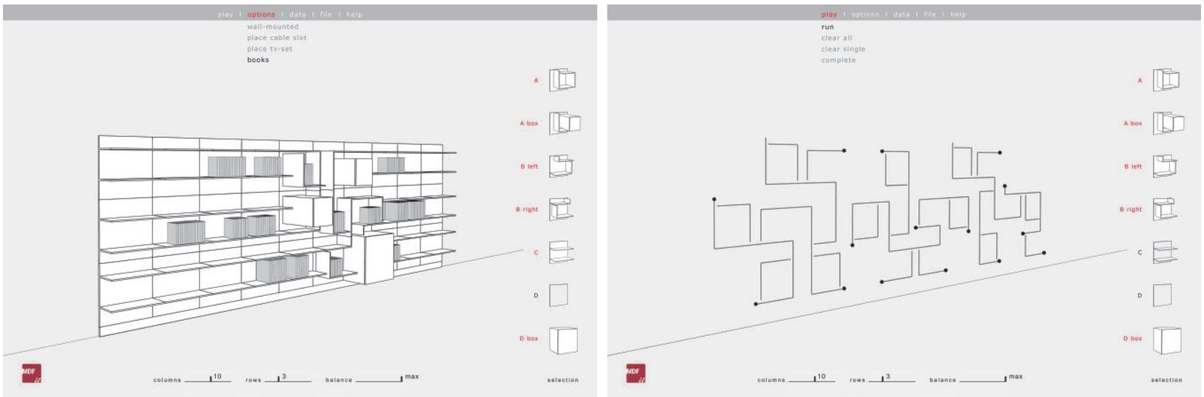


***Fig120**. VITA GUI with a generated state (left) and the diagrammatic visualization of the search process (right)*

As opposed to the CA models, no state transition function was required in the sequential growth model. Like in a CA, a cell surveys its topological neighbourhood but has its own agency of choice for action like an agent-based model (as those discussed in section 5.2). Unlike the general principle of the *voting rule* in CAs, a cell decides on its own future state and can even decide to undo itself (backtracking). A hybrid agency emerges between the utility of the grid cell as a mediator between existing branches and the behaviour of the growth algorithm, representing the observer. Direct interaction into the configuration by the observer provides feedback onto the growth behaviour but also gives a state to the grid cell. However, the interaction always refers to design purposes like functional connectivity or aesthetic continuity, not the topological cell state, which remains independent. The algorithm and observer share the same objective to create a geometrical continuation across a topological field. The search heuristics of the observer and the growth algorithm are epistemically associated while the topological structure of the model remains structurally autonomous. A three-way dialogue is established between observer –

algorithm + algorithm – structure, where the interaction as a computational mechanism enables this dialogue

***Fig121**. Analogue board game of VITA was produced to allow retail assistants in shops to ponder small area configurations. Its use showed how observers would always use the breadth-first search method rather than the depth-first search and thus align with the algorithmic simulation model*

VITA provides a unique case study as the topological autonomy and the cell ontology were developed independently from the growth system. The module designer Mariani inserted his design intentions separately from us as system designers. By opening up the growth process, the observer as user inserts his objectives into the configuration, which however is limited by the module and system design, and the design constraints (although even those can be overridden). Thus, the system was always designed as an associative structure, not as an automatic design simulation to test a designer's assumptions. The system contains two aspects: the algorithmic growth model and the interaction modes. In order to guide the use of the system, it was paramount to design the interaction and interface in such a way that the observer on the one hand understood the opportunities for driving the configuration but on the other hand retain his attention by not allowing too much interaction. The design and development of the modules and algorithms lasted approximately nine months, while the interaction and interface design lasted twelve months.

5.4 CONCLUSIONS

This chapter has introduced three algorithmic approaches to configure formal organizations in an architectural context. Bar the algorithms and their implementation which have been described above, there are concluding insights which highlight the relationship between the user as either architectural designer or client as observers and the models as well as the relationship between the models and the architectural workflow in general.

5.4.1 Observer-Associated Constraints vs System-Associated Constraints

The aim of the Object Configurations is to illustrate how the relationship between the observer and the algorithmic model is not always single-sided but can be transformed from fully automated generation to an integrated dialogue. A relation between automation and dialogue can be observed which takes place at the level of physical constraints that define the purpose of the design simulation (performance indicators) and spatial unit types.

Input for all three sections consists of some matrix or catalogue of shapes that represent well-defined units: area schedules (unit sizes and numbers) with some proximity constraints in section one; area schedules with adjacency matrices in section two; and space partitions with adjacencies in section three. Increasingly, the explicitly fixed enclosed space with dimensional constraints becomes abstracted into definitions of its fundamental elements like wall partitions (i.e. small segments of an enclosure) and their combinatorial constraints. Those elemental units with their assembly rules encode implicit mode of use: use both in the sense of type of space and as generic occupation, although occupational use in this chapter is described through geometric rules rather than behavioural analysis. As the abstraction of spatial unit increases, performances indicators become less global and quantifiable. Optimization is only possible where explicit global targets are isolated from the behaviour of the algorithmic process such as in section 5.1, where the internal epistemic process is not linked with the optimization of externally defined purposes. The assembly models of section 5.2 are measuring their own performances for the observer and indicate whether they fulfil their local agency. The topological models of section 5.3 depend on the whole system's local consensus with the observer visually inspecting whether potentials for use are occurring via emergent spatial types. With increasing abstraction of the units and increasing integration of the observer's heuristics, a decreasing scale of the morphology is specified and potentials for use established only locally.

In his book Space is the Machine Hillier (1996) distinguished two types of spatial configurations: adjacency complexes (a-complex) and permeability complexes (p-complex) and described their hierarchical relationship as: *"An arrangement of adjacent cells, whether arrived at by aggregation or subdivision, is not a building until a pattern of permeability from one cell to the other is created within it"* (Hillier 1996, p217). GA models in section 5.1 aggregate functional spaces without specifying permeability patterns and therefore represent a-complexes. By contrast, models in section 5.3 aggregate (or grow) non-functional spaces that aim to afford connectivity and therefore represent p-complexes. In Hillier's strict definition of a- and p-complexes also the topological models of section 5.3 are a-complexes as they work on the spatial delimitations of configurations not on the connectivity graph. However, units in 5.3 implicitly encode and generate connectivity graphs through geometric cell definitions (much like Hillier's original Space Syntax model) and as such fall into the same category of p-complexes without analysing the global performances of the resulting complexes.

There appears to be a clear link between the abstraction of a spatial unit and the integration of the observer: the more a spatial unit is explicitly constrained by the observer for a specific function, the fewer interfaces between the system and observer exist. And vice versa, the less constrained a spatial unit for a specific function, the more interaction appears possible as the design heuristics rather than the design purposes align. Algorithms rather than data dependencies allow for design interaction.

This is what Donald Schön (1983) intended *with reflection-in-action* and Rittel and Webber (1973) described as observation-led *satisfaction aspiration* (see 3.3). Rittel and Webber also pointed out that *wicked problems* cannot be solved by setting quantitative targets, meaning over-constrained functional specification, but through heuristic alignment.

5.4.2 Algorithmic Representations of Heuristic Process | Epistemology of Models

Each section uses predominantly one type of algorithm in a certain design context. The design purpose of all models was a generative search to explore combinatorial options in a pre-design phase without evaluating the results for user performance. With exception of VITA, algorithmic models retain their epistemic autonomy and impose their inherent knowledge on the observer as described in 3.5.

The genetic algorithms in section 5.1 provide the observer with fast global aggregation solutions where no interaction is requested and algorithmic and observer heuristics are entirely separated. A designer needs to know the data types, their quantities and some aggregation rules (*bodyplan*), which are not ontologically isomorphic (or simply structurally mirrored) within the algorithmic representation. In fact, the algorithmic logic of evolutionary principles is invisible and unknown to the observer and represents no agency of his. GAs apply their own heuristic and generate results automatically that the observer merely inspects without considering the behavioural process (*black box*). Resulting morphologies are optimized without possibility of interference by the observer in local conditions of the configuration.

Agent-based algorithms instead are specifically open to be constrained by the observer to execute some of his agency in a digital space. Hence, the mechanism for encoding agent-based heuristics are by default mixed heuristics. Agents are mostly used to explore a design space by evaluating statistical or topological conditions in a continuous spatial field. They are rarely used in direct geometrical construction of morphologies, facilitating pre-design exploration for strategic design instead.

Cellular models such as CAs can combine topological and geometrical agencies. In chapter seven further models will be shown with occupational evaluation. They enable a synthesis between global and local performances where agencies of the algorithm and the observer are mediated via a topological field. This field is not neutral due to its topological structure yet serves as map through which the intentions and heuristics of algorithm and observer are associated. In the case of CAs, transition functions serve as mediators by mutually mapping observer rules and topological field. The knowledge that this field negotiates does not belong to either observer or generative algorithm and thus sits semi-autonomously in between as an agency of both.

6 MAPPING SPATIAL ASSOCIATIONS | SPATIAL AND COGNITIVE ANALYSIS - THE MEASURES

"Field conditions are bottom-up phenomena, defined not by overarching geometrical schemas but by intricate local connections. Form matters, but not so much the forms of things as the forms between things." (Allen 2008, p218)

Chapter Five introduced algorithmic mechanisms to generate spatial configurations from explicit constraints. Apart from visual inspection and the constraint rules, how does the observer evaluate the performance of a configuration? Up to this point, this order of development is similar to Bill Hillier's Space Syntax evolution, meaning first he postulated a generative mechanism based on algorithmic representation before elaborating spatial measures by which to evaluate configurations (then the comparison ends as Hillier and Space Syntax did not return to synthesize those two strands). Coates and Frazer as champions of algorithmic epistemology for generative design were not particularly interested in spatial analysis, and thus, only few insights about performance measures were provided at CECA. In a professional context on the other hand, questions about performance are driving design and morphologies. Those performances are mostly based on design guidance providing compliance criteria. Design guidance such as the British lead-benchmark for urban planning, ByDesign developed by the Commission for Architecture and the Built Environment (CABE 2000) for the Department of Transport, have used discursive descriptions for design objectives and key-performance indicators (KPI) such as the Design Objective 'Legibility' (being the most discursive of eight objectives): "*To promote legibility through development that provides recognisable routes, intersections and landmarks to help people find their way around."* (CABE 2000, p15). Legibility as a discursive KPI with no standardized evaluation metric – usually done through physical scale models, computer renderings and photographs – contrasts with regulatory design standards such as the Metric Handbook (Littlefield 2008), which describe the performance of spaces by their explicit metric minimum and maximum dimensions. This dichotomy aligns with the distinction between parametric specification and algorithmic representation: explicit metrics and their dependencies are equivalent to static parametric dimensioning (and their dependencies), while discursive objectives encapsulate implicit dynamic performances. Those dynamics represent spatial affordances such as the user's cognitive organization of space and the subsequent choice of action. Spatial cognition in relation to the built environment takes place in movement spaces that are under-constrained by regulation and weakly programmed by organizations[40] (Hillier 1996, p197). But in reality they are full of information (spatial data and performances) that are not specified in design guidance through quantities but through discursive qualities. Weakly-programmed spaces in architectural and urban design mostly refer to semi-public un-programmed areas (Derix and Izaki 2013, p45) where intensities change over time due to generic occupation, invoking Hillier's p-complexes (1996) consisting of interfacing spaces of

---

[40] In architectural computation, *under-constrained* (Fleming 1992), *ill-defined* (Eastman 1969) or *wicked-problems* (Rittel and Webber 1973) refer to combinatorial problems, impossible to solve through an algorithmic solution path.

types C and D (see 2.7). To access their qualities for assessment, variable correlations need to be established and evaluated that relate the geometry of environments to the interstitial field where human cognition and behaviour unfolds.

Structure

This chapter will distinguish three models to quantify field conditions: maps, graphs and networks. In computational applications, most mappings are conducted on a discretized field[41], where each discrete location calculates an intensity value for itself by mapping all other discrete locations. Here, those values mostly represent some perceptual dimension. Graphs on the other hand represent the result of some calculation on a discretized geometry (i.e. a geometry broken down into some elemental parts like line segments) and are here mostly used to indicate diagrams for potential actions. Networks represent a higher dimensional abstraction by comparing and classifying map values and graph diagrams. This makes it possible to operate directly on the internal cognitive structure of perception rather than the field itself. Therefore, the structure of this chapter does not directly follow the global-local, observer-directed to observer-integrated logic but an increasing cognitive penetration of the association between space and people, first from observer-external maps of perceptual potentials, second to observer-simulating behavioural diagrams and thirdly the mapping of observer-internal cognitive states.

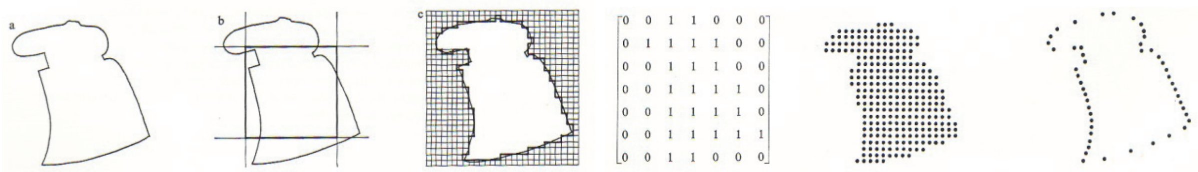


Fig122. *March and Steadman (1971, p178-9: numeric approximations of a shape by quadrat representations in increasing resolutions for best fit of perimeter (first to third images); and by a 7x7 boolean matrix (fourth image) and a higher resolution line-printer (similar to boolean) representation (fifth) and finally a raster scan representation (sixth) that provides a vector definition of the shape*

The three models of associating field conditions are not direct empirical observations from an analogue field, i.e. building or urban site, as traditionally done in architectural and urban design. They mostly represent second-order abstractions and therefore maps or diagrams of abstractions such as architectural models or urban plans. Hence, all mappings and diagrams of perceptual conditions and action affordances need to be understood as potentials of the abstractions that is provided by the input. Those maps should not be confused with real environments that they represent, heeding Jean Baudrillard's observations on the use of digital simulation to invert the mapping principle: the map precedes the territory that it is meant to map (Corner 1999, p222). In other words, too often maps of simulated conditions are mistaken for reality.

[41] Planar fields for mapping geographic territories are mostly represented via orthogonal grids due to array indexing, i.e. a two dimensional index (i,j) produces in the simplest case a square; and subdivisions to approximate resolutions that cover the minimum size of elements in a field such as fractal decomposition or quad-tree partitioning (Kodicek 2005)

6.1 MAPS | FIELD-BOUND PERCEPTION

"It follows that since points of potential observation are contiguous so then is the information available spread throughout space in a field-like way." (Benedikt 1979, p48)

Mapping perceptual properties in an environment has a long tradition in urban planning. Kevin Lynch's 'mental maps' (1960) of American cities raise perceived dimensions directly from the territory via interviews. William Whyte (1980) mapped behaviours associated to properties of public urban spaces into grids via time-lapse photography in his Street Life project. Both mappings have in common that qualitative data is gained directly at the spatial location of the territory. Whyte however subdivides space and abstracts behaviour into discrete positions and attributes quantitative values. Because of the regular subdivision, each position value geometrically represents a succession from neighbouring positions. Hence, values in a discretized map still represent a continuous field (unlike most topological representations like graphs or networks).

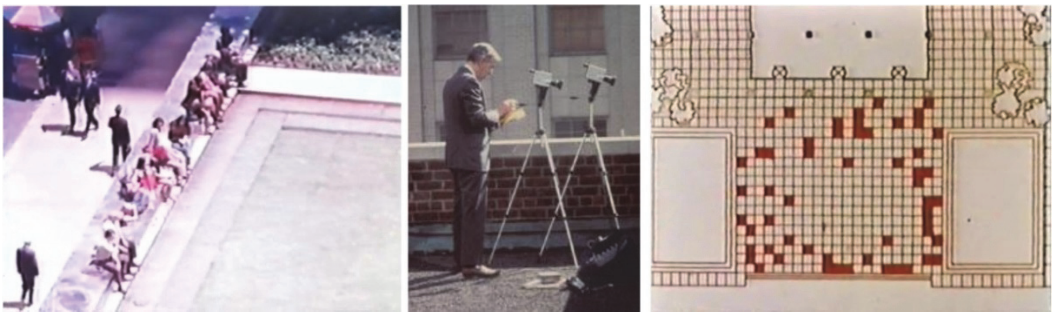


Fig123*. William Whyte, 1979: Street Life Project, mapping public spaces in New York using time-lapse photography to produce discrete activity maps*

Whyte's mapping was passive where positions did not calculate their own values but were given a state by external observation. In computation, each position actively calculates its own value from its context through a function of environmental probing. A computational map therefore represents a local array of performances, which the observer can globally differentiate (as the position does not 'know' anything about the rest of the map). Because the discrete position values are directly analogue to the territory (or the abstraction used), it cannot be transformed or instrumentalized for scenario planning (unless some reverse projection function exists which adapts the properties of the territory to a transformed map). Hence, discrete maps are used to inform global observations for strategic design decisions. At CDR, maps have mainly been used to calculate positional perceptual properties that approximate discursive qualities constituting KPIs.

6.1.1 Continuous Perception | Structure-bound

At CECA computational or rather algorithm-based mapping was explored via agent-based explorations. This shows that concepts of spatial knowledge, regarding epistemic properties of algorithms, preferred space and iterative process to be a continuous field experience, simulating embodied experience rather than analytical reduction. The most adequate algorithm for such as an approach provides the continuous sampling of location-based agents who iteratively sample successive positions.

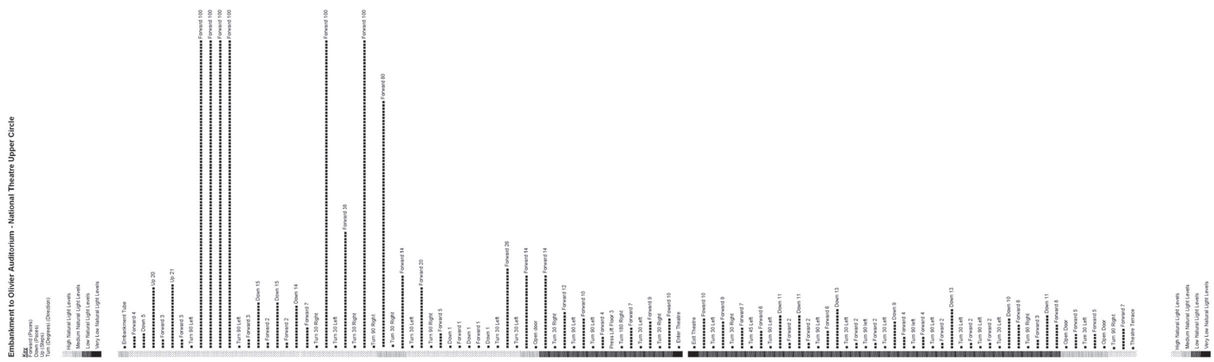


***Fig124**. Josie Elt at CECA, 2007: route experience mapping light qualities (grey scale) and turns for no. of steps (no. of vertical boxes)*

The agency of an agent in mapping consisted of two methods of probing for the observer: the condition at a position and the perception of the surrounding environment. The first method constitutes a sensing of a local condition at a discretized position (in plane or space) and depositing the perceived intensity as a normalized value at the occupied position, first trialled by Grey Walter (1950).

At CECA, Pablo Miranda in 1999 and Jasminder Parvin in 2002 replicated Walter's analogue agents with structural additions proposed by Valentino Braitenberg (1984) to explore the correlation between physically determined perception and observed environmental patterns. In Vehicles Braitenberg (1984) theorized on embodied perception, suggesting that the structural configuration of sensors and actuation of motors drive environmental cognition. Only a sub-space of the total available field that the agents are situated in is mapped due to both the structural relationship between sensors and motors and the light threshold set for the sensors (Fig125). Spaces are mapped that are isomorphic to the agents' structural configuration.

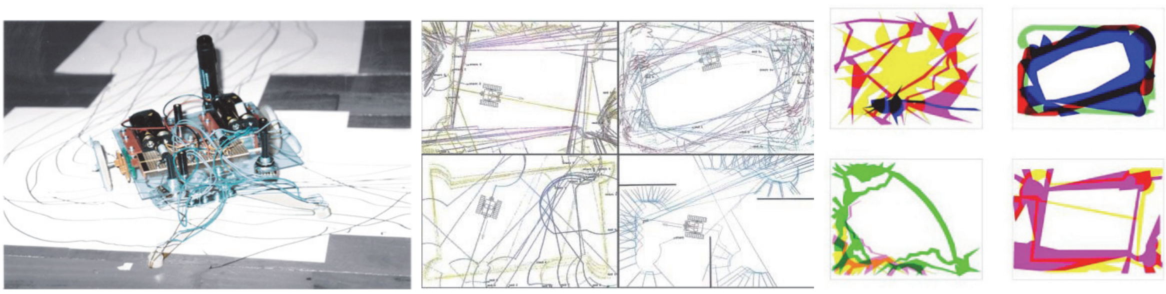


***Fig125**. Pablo Miranda re-producing Walter's Elmar and Elsie light-sensitive robots, 1999 (left); (middle) Jasminder Parvin simulating Elmar and Elsie's structure and behaviour algorithmically to visualize occupied space; (right) mappings of the amount of space perceived and occupied by Jasminder's robot agents*

The second probing method was mostly explored with computational agents. The perception of the environment again stems from Kevin Lynch's (1960) work on spatial orientation where he introduced the notion of well-structured environments by identity through elements such as landmarks, distinguishable routes and frontage rhythms as basic structures for way-finding (Lynch 1960). This was later formalized by researchers such as Romedi Passini (Arthur and Passini 1992) and Ruth Conroy-Dalton (2001) who identified and quantified parameters for navigational choices.

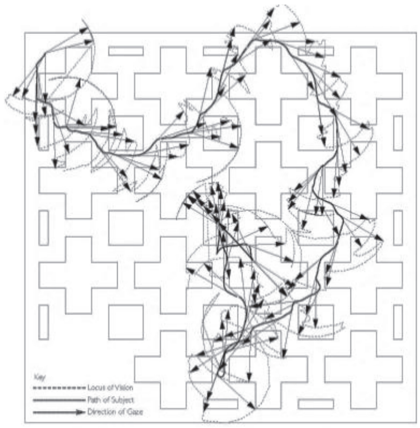


Fig126. *Ruth Conroy-Dalton: spatial navigation in immersive virtual environments (Conroy-Dalton 2001)*

Like the analogue robot agents, this mapping technique shows how an environment supports visual perception and hence affords movement behaviours. Both models demonstrate that a perceptual model requires the understanding of two structural ontologies: the configuration of the perceiver and the configuration of the environment.

VISUAL AGENCY

These two methods of location sampling and context probing produced sparse mappings and particularly for a professional context it was beneficial to produce consistent performance values to allow the evaluation across a complete site. The CECA agent configuration was modified to produce perceptual support maps for live projects. During the competition for the Museum of Modern Art in Warsaw in 2007 two types of agents were developed to map perceptual conditions of the site. Pablo Miranda developed an agent-based visual flow mapping using an adaptation of the optical flow algorithm by Beauchemin and Barron (1995). The author developed a visual access mapping, which also provided the basis for an integrated generative method.

For this competition, the original task was to map the accessibility conditions of the site, which was located within the same urban block as the Soviet Palace of Culture. The rectangular site was only accessible by four subway passages at each corner with the Palace obstructing the view from the subway exits to the potential museum entrance. The analysis of visibility and accessibility conditions were meant to inform the massing design. The maximum envelope was given by the client from which to derive the massing.

The Visual Flow agents mapping tried to establish a map of visual conditions to indicate the direction of view and movement of visitors on site. Visual Flow refers to the technique of Optical Flow, itself a branch of computer vision. Optical Flow maps visual change between an observer and the observed (the context) by analysing changes of intensities within a discretized visual field of the viewer. To analyse change from time *t* to *t+1*, any quantifiable data can be drawn on to calculate the differential and the discretized space can represent many types such as geographical position, pixel or voxel. If an object is moving, adjacent locations will change their differential and a directional field is established, indicating either the observer's or the object's movement path. The intensity of change can be measured by the amount that a parameter in the discretized location varies between frames (Beauchemin and Barron 1995).

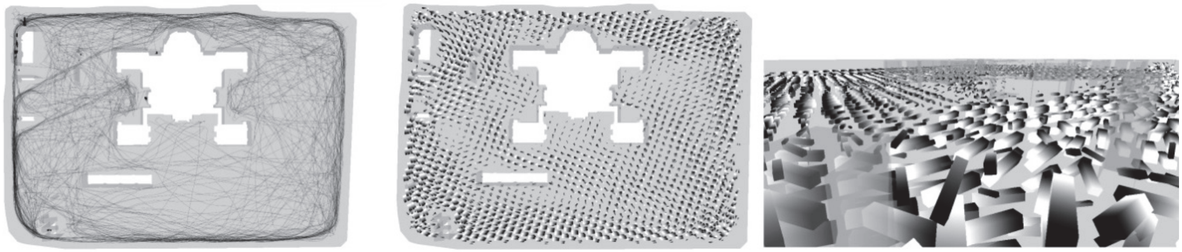


***Fig127**. Warsaw MoMA, 2007: left, traces of agents; middle, visual flows showing 'blind spots' on site or strong visual and movement areas; right, grid of intensities across site visualized as direction vectors rendered as transformed boxes*

Visual flow agents were released randomly within the site and moved with a long straight heading probe until encountering an obstacle. The incident angle between obstacle and agent would determine the future heading of the agent. The site was discretized into regular grid patches (positions), which recorded movement information of agents such as number, speed and orientation. The information was stored as a trace vector, which would age and decay dynamically (number of steps), helping the patches to determine the age of the event. If multiple agents crossed a patch during the decay time, the intensities were summed up. The parameters of intensities were averaged over time to reveal the general through-traffic, speed and orientation with each location. Proximity to obstacles would slow and deviate agents, which when relayed into the patches, would approximate search behaviours at such locations. After each loop, the intensity values were diffused into a topological Moore neighbourhood to relax the map and reduce local statistical irregularities. The map emerges over time as a result of agent traces producing an indicative diagram of avoidance movement and visual access conditions across the site.

Visual Access agents used the three-directional probing structure discussed above for obstacle avoidance and evaluated the possibility of seeing the museum entrance within their FOV at each location. Several mappings were conducted that showed the possibility of seeing the museum entrance at each location, varying site exit strategies (number and location), presence of the provided museum envelope (as current and planned scenarios). Again a diffusion function distributed the visual access values of each location to Moore neighbourhoods mapping averages across the site. The resulting maps not only showed visual access conditions but implicit weighted connectivity graphs emerged revealing potential links between site access points and museum entrance locations.

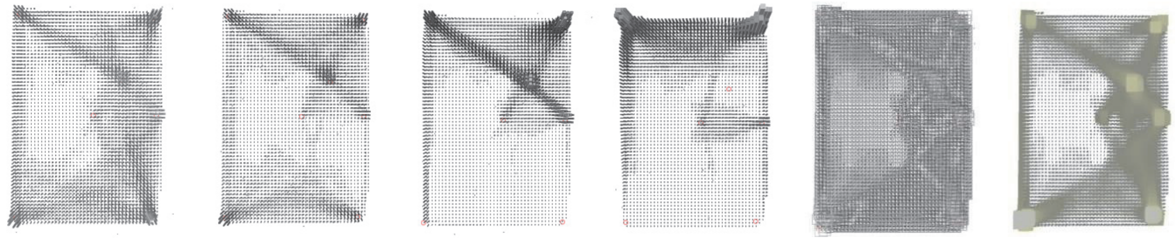
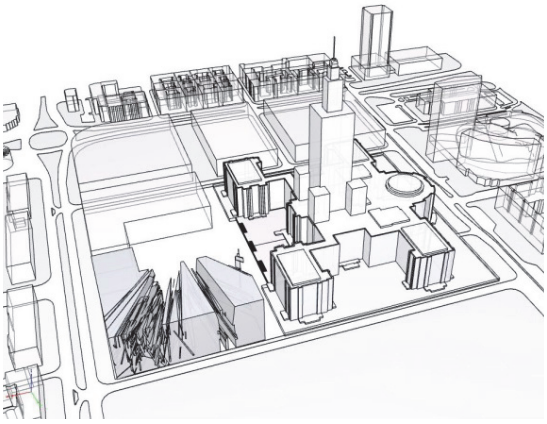


***Fig128**. Warsaw MoMA, 2007: left, multiple scenarios with varyingly activated access points; right, summary of activation by visual access and the connectivity graph between access points*

Visual Access agents also operated directly on the maximum prescribed envelope to illustrate disparate access conditions by cutting away at the envelope highlighting problematic directions of occlusion for way-finding. The transformed envelope represented a map in its own right, which strongly informed the final massing strategy developed DavisBrodyBond-Aedas in New York. The envelope was subdivided into spatial bands which were offset between themselves to allow visual permeability through the building massing and increase visual access across the site.



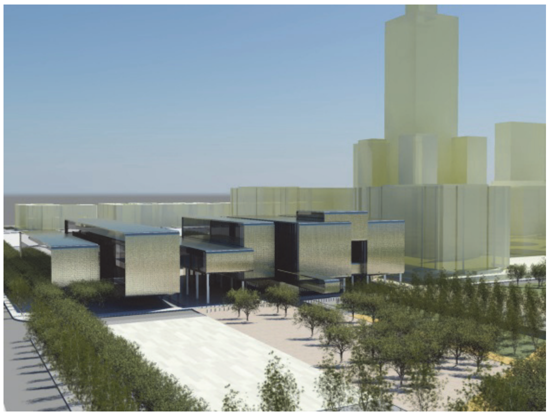


Fig129. *Warsaw MoMA, 2007: left, maximum prescribed envelope eroded by visual links; right, final design strategy of subdivided spatial bands to allow visual permeability across site*

The MoMA design shows that the agents' actions are driven by the information available in the environment as first suggested by Kevin Lynch. The psychologist James Gibson (1950) anticipated the optical flow concept by theorizing a vector environment that guides the visual perception of the occupant, called the Ambient Optical Array, attributing the observer a participatory role as decoding instrument of environmental information. Walter and Braitenberg's experiments elaborated observer perception to simple mechanical configurations by visualizing their behavioural responses to environmental intensities. Braitenberg particularly wanted to decode 'emotional' cognition such as 'fear', 'aggression' and 'love' as an effect of the spatial-user-observer correlation (Braitenberg 1984, pp6-14). Similar to Braitenberg, Craig Reynold categorized agents into semantic or emotional categories (Reynolds 1999). Eventually, a generalization of some behavioural processes for a generic occupation model was developed, called People Movement, to enable the evaluation of live projects during early design stages.

MOVEMENT AGENCY

The People Movement plug-in to AutoCAD was meant to be a generalization of behaviours for generic occupation, yet weighable between some selected behaviour properties. The simulation was meant to evaluate generic conditions such as flows, movement bottlenecks (generally called *queuing*), cognitive weaknesses of the layout (impaired way-finding) and timetabling issues (school pupils collisions in room exchanges for example). As opposed to commercial pedestrian simulation software, agents were meant to 'perceive' the environment during processing and 'act' socially. This *embodiment* or *situated perception* principle for simulation postulated by Valentino Braitenberg (1984) was to help approximate generic occupation.

Commercial software in 2004/5, for example Exodus[42] or Steps[43] relied on pre-processing the environment by discretizing space and attributing movement values to each position such as proximity to available targets etc. At run-time, particles would query each location for the optimal next move towards selected targets[44], creating essentially a deterministic simulation (with some randomness), which would not be able to reveal any cognitive properties of a layout other than the proximity grid maps containing the directional choice of movement for the particles.

Several tests were conducted with various general movement behaviour algorithms such as the

- Basic three probe turtle/ boid agent described above as the CECA agent
- Selection of steering behaviours by Craig Reynolds (Reynolds 1999)
- Ant-Colony natural optimization (Bonabeau et al. 1999, p31ff)

The basic CECA agent based on Reynolds' boids, provided the general propagation algorithm (Reynolds 1987). Reynolds' steering behaviours helped to implement some additional contextual as well as social awareness such as

- Seek: a target search with direction and velocity regulation (target needs to be visible by probe)
- Obstacle Avoidance: using the basic CECA agent deviation model rather than Reynolds' sphere intersection procedure
- Unaligned Collision Avoidance: preventing agents from positional encounter on next moves
- Flocking: for social behaviour such as group cohesion and alignment of directions
- Leader Following: developed by author independently from Reynolds' method for quasi-communication between agents if a target is found

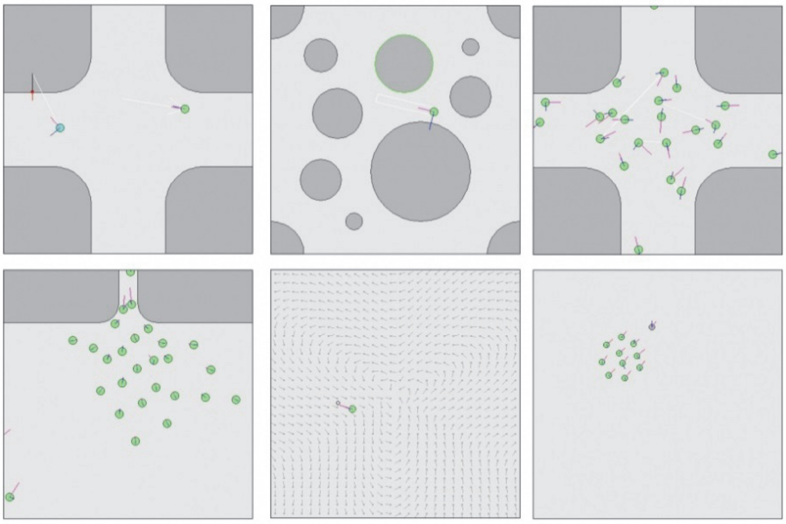


***Fig130**. Reynolds, 1999: six steering behaviours from top left: containment, obstacle avoidance, unaligned collision avoidance, queuing, flow-field following, leader following*

[42] http://fseg.gre.ac.uk/exodus, accessed 05.08.2014
[43] http://www.steps.mottmac.com, accessed 05.08.2014
[44] Essentially a *hill-climbing* algorithm

Additionally, a modified ant colony optimization (ACO) was developed to calibrate the model towards an analogy of experience of an environment as different typologies of buildings operate with varying levels of place knowledge of their users, for example hospitals assume low place knowledge of out-patients while schools assume high place knowledge of their pupils. This feature was not integrated into the final 'people movement' model.

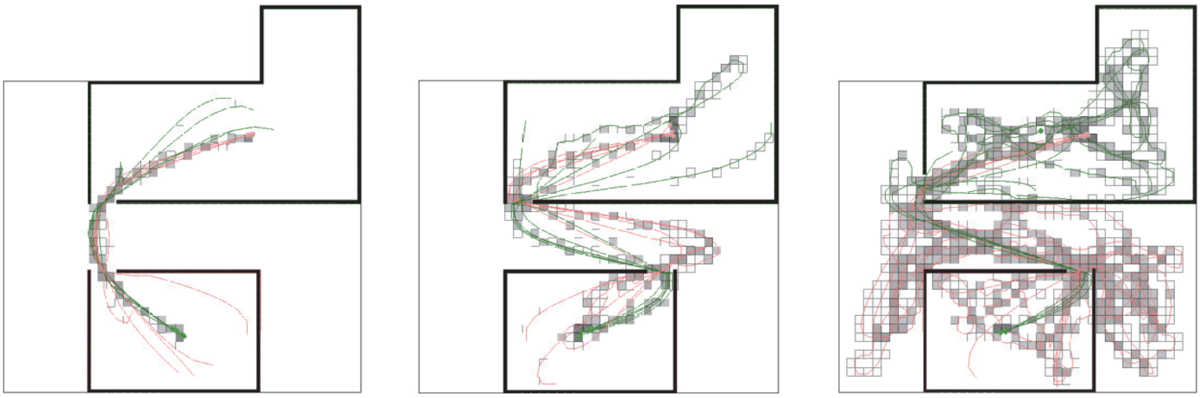


***Fig131**. People Movement model, 2006: three scenarios with different door location between two rooms showing how effectively agents without previous place knowledge manage to switch rooms*

The generalized People Movement model was applied to two live projects within Aedas from 2005-6, both secondary school buildings where scheduling and encounters matter for layout design. On Bromsgrove School, Worcestershire UK, the model was used to evaluate the entrance foyer of the new school building during scheduled room exchanges. A patch-mapping visualized possible wear-and-tear areas and allocation of toilettes around the foyer was re-considered due to minor problems with pupil encounters. Overall however some misunderstandings of the use of 'generic occupation' simulation exists in a professional context as the value of movement simulation with agents (and currently particles and deterministic maps) is usually only perceived as a means to a) design out problematic locations in egress and crowd emergency situations, b) increase efficiencies of space utilization and c) for compliance and presentation submissions. Issues of general layout based on 'way-finding' design with cognitive qualities of a spatial environment are deemed 'obvious' and reviewed through 3D models and renderings. For presentation purposes, lifelike appearances are preferred over diagrammatic spatial representations.

The People Movement prototype was developed for spatial *designing* purposes on an architectural scale where intricacies of geometric properties and perception for way-finding matter. Its many cognitive, social and spatial performance criteria are difficult to scale up to a large area with many thousands of people, particularly in emergency situations. Despite this limitation for the commercial market, its innovative potential was recognised and in 2006 as MottMcDonald's Engineering ltd commissioned the author to design the scope for the next development phase of their people movement simulation software called STEPS, which is now running on a continuous agent-based model.

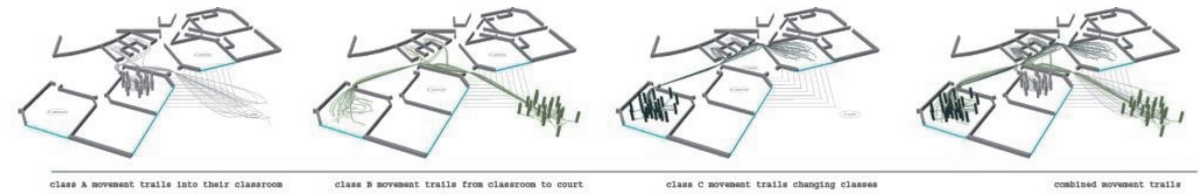


***Fig132**. People Movement, 2005: three classes moving between locations during a break for planned Bromsgrove academy, showing each class' movement traces and (right) all three classes' movements*

6.1.2 Discrete Perception | Territory-bound

A second stage of the People Movement simulation has been conducted by Asmund Izaki of CDR in 2010 based on the leaner stand-alone programming environment Eclipse for Java, which increases speed for real-time visualization and allows the up-scaling of numbers of simulated agents. Because the purpose for professional evaluation of movement behaviour often aims at efficiencies rather than cognitive qualities of space, a discretization appeared feasible to map all positions in plan rather than only used spaces. Like early commercial software, we separated a pre-processing of the configuration from the simulation of the agents' behaviours, without however committing the previous commercial tautology of agents being deterministically guided by the pre-processed environmental information.

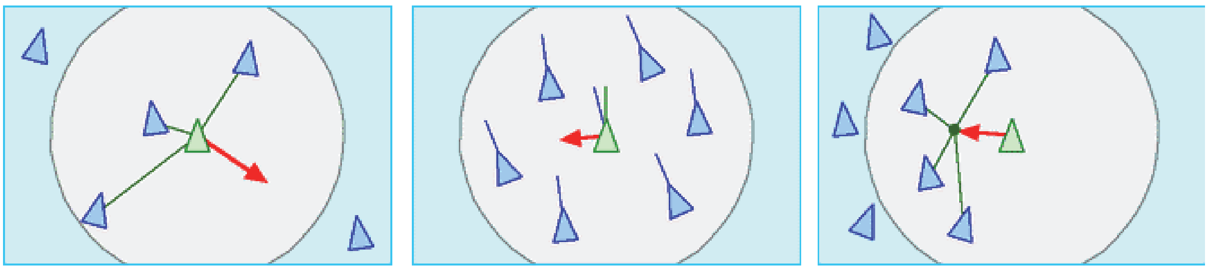


***Fig133**. Reynolds, 1999: three principal flocking behaviours, left to right: separation, alignment and cohesion*

The pre-processing stage consisted of the calculation of a radially-constrained visibility network where each discrete position would be connected to all other visible positions (generally called the 'visibility graph' - see chapter 4.2). The output is a distance field containing nodes with a probability for choice of movement towards a target like a hill-climber algorithm. Agents encoded four of Craig Reynolds' steering behaviours (containment, un-aligned collision avoidance, flow field following and separation) and mediate the contextual geometry, social interaction and path probabilities simultaneously in run-time. The visualization shows agents navigating from set origin to destination points (OD), the traces of their movement, encounter locations with number of collisions and gate counts at user-defined locations such as corridors. Due to its real-time rendering of each agent step and choice of movement, an empathetic link is created between observer and model as the observer believes to understand the decision made by the agents. The mapping technique represents a transition from continuous to discrete, combining global static and local dynamic perceptual quantities.

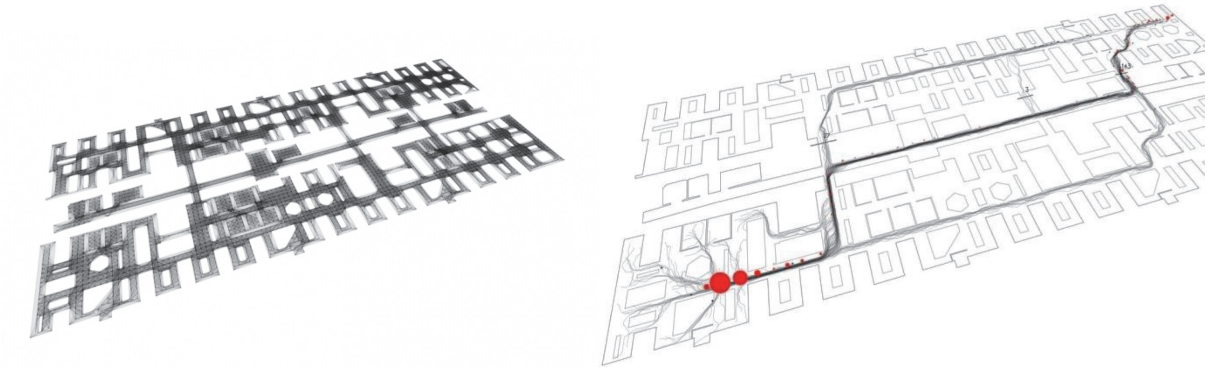


Fig134. *People Movement II, 2010: left, the visibility network constrained by a distance radius between nodes and right, agents taken choices between routes by probabilities and social interaction*

Instead of pre-processing a geometrical environment into a graph-like network, topography can be utilized directly as a choice-landscape if its geometric representation resembles a terrain mesh. A geometric mesh is built like a graph network with nodes and edges. Hence, geometric values can be calculated along edges. In the simplest case, an edge between two nodes embodies spatial properties of proximity, orientation and slope. If a graph is considered a sub-set of a network, then a mesh does not strictly represent a graph, as both of those examples operate directly on the territory, not on an abstracted representation of this territory (clearly the mesh or the visibility graph in People Movement II are abstractions but they are analogous to the geometric locations provided by the first territorial abstraction).

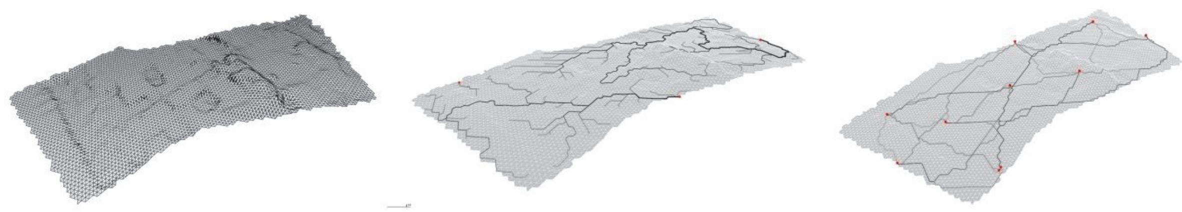


Fig135. *Greater Noida, 2008: left, edges of mesh weighted by slope; middle, separate origin-destination paths by slope across terrain; and right, slope-weighted route network equalling origin and destination points*

AVOIDANCE AGENCY

In 2008, CDR developed a movement analysis prototype for a mixed-use masterplan in India, a Special Economic Zone (SEZ) for Greater Noida. Access pattern from secured entrance locations into the site were to determine the allocation of land-uses. In the absence of meaningful contextual drivers (the site was barren and outside a town), it was suggested to use the site topography as drivers for land-use allocation in relation to site access points, because heat and therefore resistance to movement could be considered the only behavioural driver. The edges of the terrain mesh provided the input to calculating the shortest routes on the mesh between an OD pair. A Dijkstra route algorithm was coupled with a hill-climbing algorithm to allow for the weighting between shortest and flattest routes. Dijkstra shortest routes are calculated on a network of edges by breadth-first search that, similar to agent-based logic, constructs edge routes into a continuous path, which can be evaluated. While the choice between two nodes in a standard Dijkstra algorithm is based on topologic depth or geometrical distance, for Noida we used a hill-climber to choose the edge with least slope (height differential). Slope and directness (shortest vs direct) could be weighted via sliders, which resulted in the mapping of the topography according to topographic and mesh properties, producing a map of the energy cost to traverse the territory. When multiple points were considered as equally OD pairs, a route network would emerge where shared edges emerged offering opportunities for movement across the territory. This type of network would allow for multi-modal transport simulations where similar behaviours between some modes occur such as pedestrians and bikes, while cars navigate the field differently. Section 5.3 will explain the land-use allocation in conjunction with topographic network mapping.

***Fig136**. Greater Noida, 2008: weighting routes from all mesh nodes to a destination access point from left = direct routes + no slopes ,to right = shortest routes + minimal slopes*

REACHING AGENCY | VISIBILITY GRAPH VS VISIBILITY MESH

The concept of the Visibility Graph in urban design is based on the notion of sightlines in a planar environment while originally, visibility graphs were developed for robotic movement prediction of unobstructed corridors called 'motion planning' (O'Rourke, 1994). The visibility graph is calculated by connecting all intervisible vertices of polygons (usually in a 2 dimensional plane) through an edge. This representation is a high-level abstraction removing the discretization from the underlying geographic territory. The field between buildings (or other geometric features in plan) can be filled with regularly spaced grid positions that together with the polygon vertices can be transformed into a mesh. Such a planar mesh provides the basis to produce a dense visibility graph for analysis as done in the previous project. But instead of measuring slopes or shortest routes between OD pairs, general catchments can be analysed, which would not be possible on a standard graph as a sub-set of the field only representing obstacles, because no open space positions are sampled for a generalization of the field. Hence, a distinction can be made between a 'visibility graph' and a 'visibility mesh' , where the graph represents the field through visual obstacles and the mesh represents the field through an integration of open space and obstacles. The purpose of such a visibility mesh is to map the 'reachability' in a field according to different behavioural measures instead of exact routes as done with a graph.

LB Newham - Public Transport Accessibility Levels

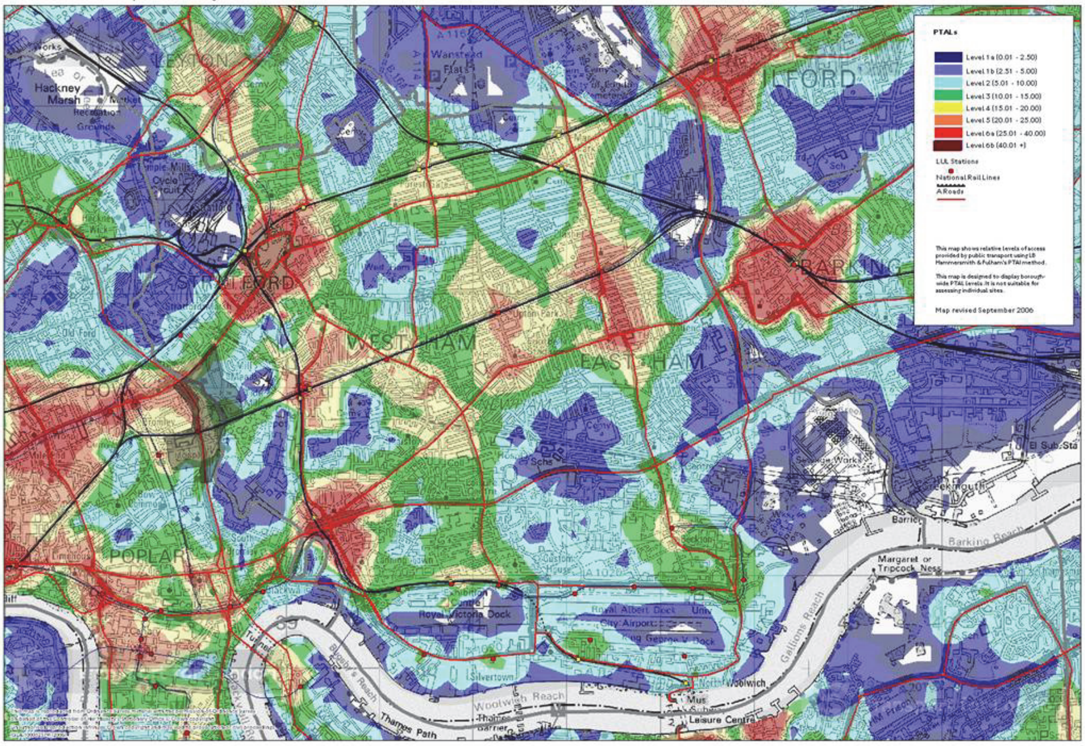


***Fig137**. Public Transport Access Levels (PTALs): map showing access levels by pedestrian distance to transport nodes. Six levels with internal sub-levels exist to provide information about density or land-use*

From 2007 to 2008, CDR produced a series of urban design simulations for a UK-government funded project called *Smart Solutions for Spatial Planning* (SSSP) (Derix, 2012). The project aimed to provide knowledge transfer between professional and academic silos in urban planning and a series of workshops were organized in collaboration with Paul Coates at UEL, to exchange statutory and heuristic knowledge of urban planners with computational designers. The workshops revealed that many urban planning decisions are founded on accessibility criteria, which in turn inform aspects such as density and land-use allocation. The key analysis used by planners was the Public Transport Access Level maps (PTAL) developed by London Borough of Hammersmith and Fulham in 1992, which evaluates distances to public transport nodes and their frequency (Transport for London, 2010). But the main indicator is the accessibility catchment and its metric distance, which as standard is calculated as pedestrian walking time.

Since it was felt by our funding partners that catchment levels were such decisive criteria for urban planning, CDR decided to develop an interactive simulation for mapping catchment levels. But as PTALs were originally developed by and for transport planners, not urban planners, it was decided that the key objective of a new simulation should be the focus on a set of access measures without the transport frequency. The simulation application discussed here called Fields was written in Java by Asmund Izaki and has been updated for the Masdar Zero Carbon City mixed-use neighbourhood MIST 340 in 2009.

Fields takes as input a DXF planar geometry file with closed polyline polygons for obstacles. Access points can be seeded inside the DXF file or interactively through the GUI. A mesh is produced from the polygon vertices and the interpolated field positions with a user-defined grid spacing. From this mesh a visibility graph is generated that allows for a continuous mapping of open space distances to all access points from each position. Three types of distances between all mesh nodes (positions) and the provided access points are computed: metric, topological and angular. The three distance analysis modes correlate to different user behaviours (Turner 2000; Conroy-Dalton 2001; Turner 2009), namely

- metric computes the shortest distances on the mesh and represents users with good local knowledge such as residents
- topologic computes the least-turns distances on the mesh and represents users with little local knowledge such as tourists or hospital out-patients
- angular computes cumulative angles (or least deviation) on the mesh and represents a variety of users such as tourists or cyclists

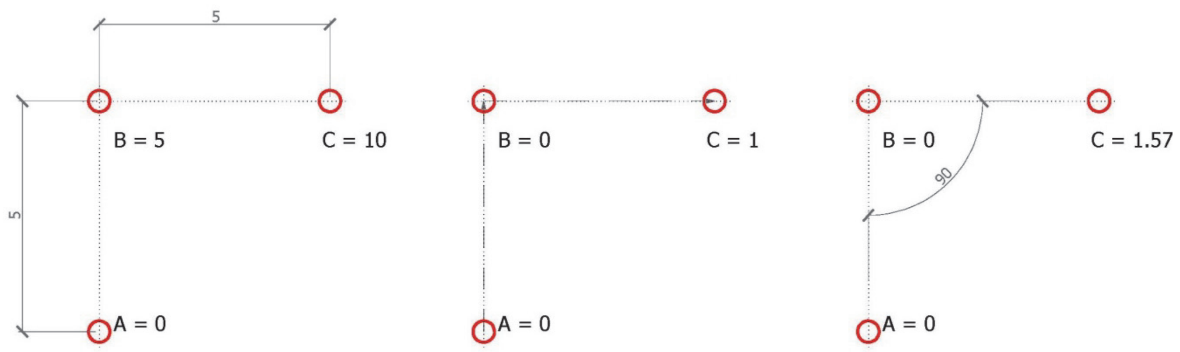


Fig138. *Three analysis modes for distance from node A to C via B: left, metric distance equal 10 units; middle, topologic distance equals 1 turn; right, angular distance equals 1.57 radians (90 °)*

The distances are computed using the Dijkstra path algorithm with different weightings. All nodes on the mesh calculate the three distances with a breadth-first search outwards from the access points, giving each node a value for each distance. The user can set thresholds for each distance mode to visualize various catchments. The catchments can be visualized as colour gradient maps or outlines with number of levels. The access direction for each node can be rendered by the user as direction vectors, giving an indication of flows through the urban field.

The Fields mappings show how users are represented via behavioural definitions utilizing geometric abstractions. Each map offers a reading for the observer specific to the strategic planning contexts. As suggested by Hillier and colleagues, metric shortest routes analysis represents local movement patterns better while angular and topological analysis represents global movement patterns. Local refers to people with high knowledge of the local context (street or movement network) and global refers to people with little local context knowledge and people with high global context knowledge, i.e. long distance journeys across larger cities (Hillier et al. 2010).

The metric shortest distance informs strategies for standard access patterns of local urban and building users who have high knowledge of their context. In Fields, user behaviour was assumed to be the choice of the shortest metric route from anywhere to the access point. In an urban context, this partitions the site into near-convex areas with no definition of hierarchies of streets but a definition of proximity through

desire lines towards the access points where footfall for specific land-uses such as groceries or news agents is high[45]. This type of user engages less the urban morphology than movement from memory (Montello 1991).
Topological and angular routes on the other hand support planning strategies for non-local users like tourists at a neighbourhood scale and global movement such as vehicular traffic (Turner 2009). For an urban impact study commissioned by the CrossRail consortium in 2009 to analyse potential footfall patterns within the catchment of London Whitechapel station, both users were mapped: locals at rush-hour and tourists between rush-hours for the market on the high street and out-patients to the Royal London hospital. Two patterns of potential footfall movement emerged that helped allocate street crossings and land uses. As the description *least-turns* routes for topological analysis makes self-evident, a route is calculated with least nodal turning, or depth. In an orthogonal grid this is also called the Manhattan distance. When degrees of angles between mesh edges are summed up, the angular distance calculates routes with least cumulative angle between an OD pair. When moving at higher speeds with larger turning circles such as vehicles and bikes, this analysis is more pertinent, as it provides an indication of least interrupted journeys.

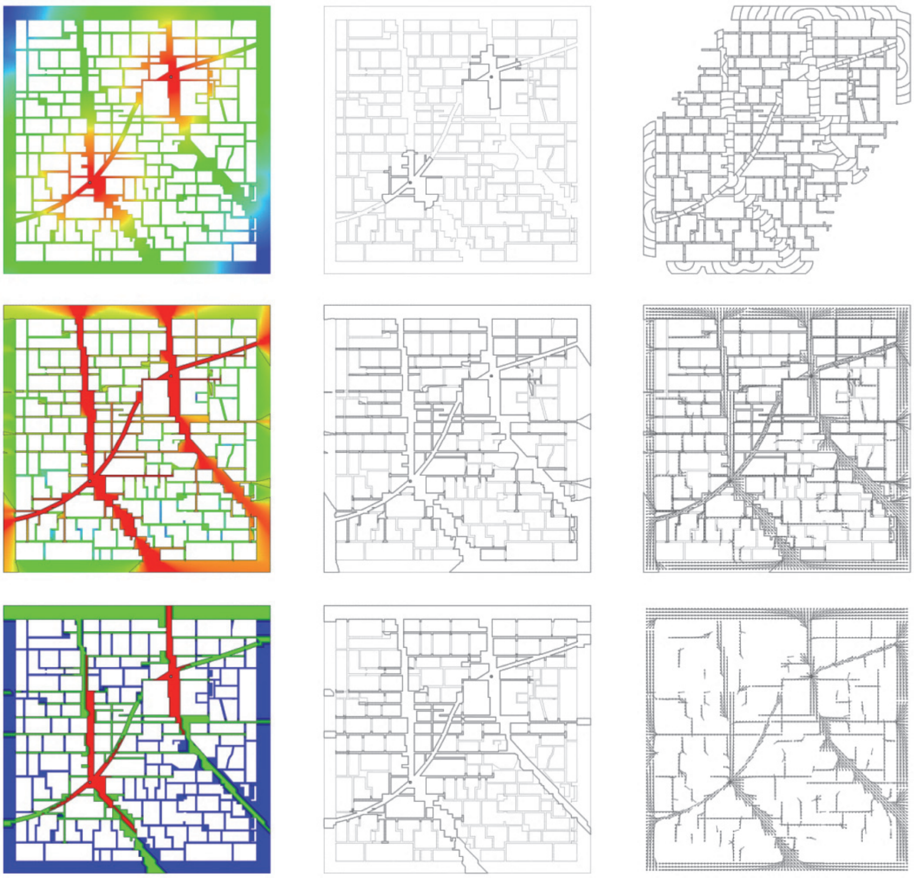


***Fig139**. The Fields tool used on the Masdar Zero Carbon City masterplan, showing three types of distances to the Light Railway stations: metric 200 meters (top); angular 90°(middle) and least-turns 2 turns(bottom)*

[45] Much research exists validating correlations between route algorithms and network measures such as centralities. An average maximum correlation between land-use (particularly retail and commercial) and network and flow properties of up to 70% has repeatedly been found (Chiaradia et al 2009; Porta et al 2009, Stonor 2014).

EXPOSURE AGENCY | OVERVIEW

Although the underlying representation is based on visibility properties, most models in this section have dealt with the effect on movement where visibility conditions produced decision maps for movement behaviour. Architectural practice deals with many visibility conditions when planning a building or urban configuration. In 2004 syntactical visibility analysis existed only as academic research and as an evaluation service for two dimensional by the Space Syntax limited and Intelligent Space partnership[46]. Both companies apply versions of Alasdair Turner's Depthmap software (Turner 2001), which integrates Visibility Graph Analysis (VGA –4.2), as an external consultancy service with static reports as output. The first development of the author at Aedas as CDR constituted the development of an interactive VGA plug-in to AutoCAD 2005 (R19), called OverView, so that all architects could integrate visibility analysis into the design process at any stage. The purpose of the development was initially firmly placed in the educational sector to support the design of open spaces in schools where bullying can be problematic. Traditionally, attempts were made to limit bullying by active surveillance of the school breaks by high numbers of teaching staff placed in open spaces. Nowadays however, intervisible spaces are preferred to allow for passive supervision (or also *natural surveillance*). Passive supervision aims to visually integrate spaces so as to design out secluded areas so that people can monitor each other (ACPO 2014). In schools this helps reduce bullying and also reduce costs for (active) surveillance.

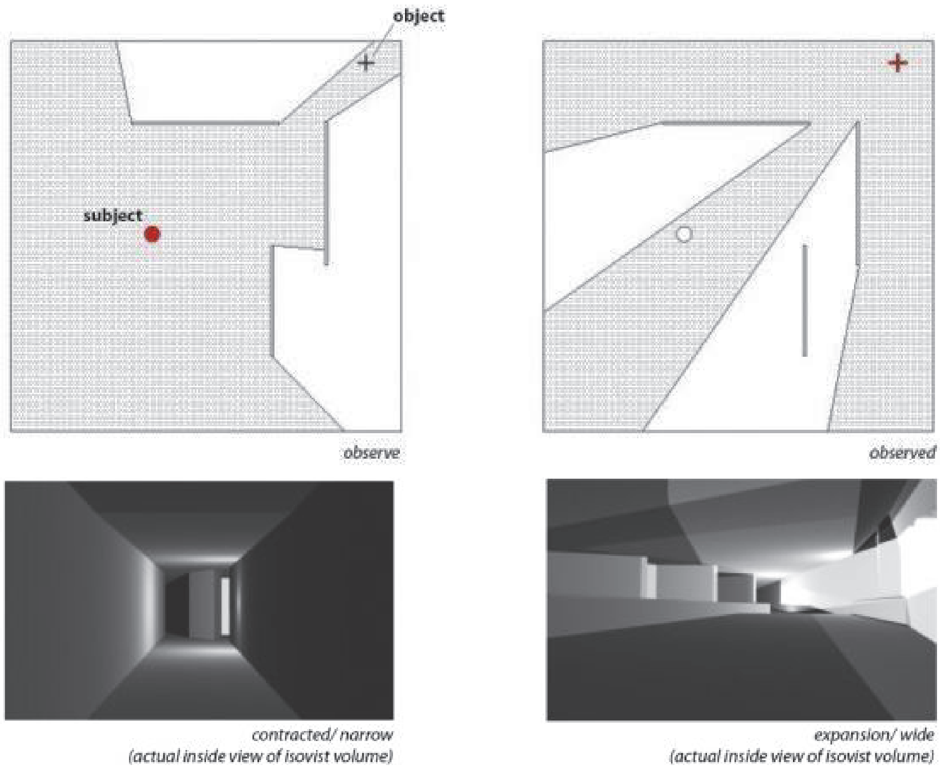


Fig140. *Exposure - observing and observed: images taken from the September 11th Memorial Museum Visual Analysis report, 2007 (CDR, 2007), showing (top) the differing placings within a viewshed of two intervisible locations and (below) the visual impressions of the viewsheds where a viewing subject would be framed*

[46] Both spin-offs from University College London (UCL) Bartlett Graduate School: Space Syntax limited co-founded by Bill Hillier and Alan Penn, and Intelligent Space founded by Jake Desylllas who graduated with a PhD from the UCL Bartlett and first worked at Space Syntax limited as a director.

The notion of passive supervision introduces the concept of visual exposure and highlights that visibility is not only active perception but also an undergone sensation, meaning a cognitive impact of the environment onto a perceiving user. During the development of the first ever three-dimensional viewshed analysis for the National September 11th Memorial Museum in New York (see below), the author speculated on the mutuality of intervisible locations, coming to the conclusion that despite the mutual unobstructed view, the perception of each other depends on the setting within the viewshed, especially the subject's framing within the FOV (Fig140) (Derix et al. 2008). Daniel Koch (2012) later called this the 'logic of the mannequin' where a viewshed is not only an active position of viewing but also each viewing location is included in somebody else's viewshed.

As discussed in chapter 3.4, the analytical viewshed analysis was introduced by Michael Benedikt (1979) as the concept of the Isovist. Being a mathematical description, Benedikt's isovist did not take account of the seen space but reduces views to perimeter lines. Alasdair Turner of UCL picked up on this as a problem for architects for two reasons: firstly, the geometric radial generation does not allow for the evaluation of the space within the viewshed and secondly, Benedikt's measures are therefore geometrical but not spatial (Turner el al 2001, p. 104). Turner therefore proposed a new generation methodology based on a discretizing a layout plan into dense grids of points that evaluate each other. This leads to an underlying structure of connections within the grid that shares properties of a graph and can therefore be evaluated as a network, enabling the integration and evaluation of visual properties of all isovists at each location as a correlation to the layout as a whole. His developed methodology is no longer mathematical but syntactical and more general, making it easier to apply to any layout for comparative analysis. This method is called Visibility Graph Analysis (VGA – see chapter 4.2 for its measures).

The graph theoretical aspect in Turner's VGA refers mainly to the underlying structure of the calculation as a network of linked edges between grid nodes, which allows us to classify his discretization and construction of a regular field of nodes in this section of maps rather than graphs. Turner's intention was to cover a plan layout as a continuous field, making the eventual visibility graph a direct, yet discretized map of the (abstracted) territory. From his proposed measures (neighbourhood size, clustering coefficient and mean shortest path length) only the third is based on graph theory, which evaluates the topology of the isovists fields as a higher order abstraction removed from the territory (Turner et al. 2001, 115-119).

OVERVIEW IN PRACTICE

The initial development of OverView for passive supervision was based on Turner's visibility graph structure, calculating the intervisibility between all linked nodes within a grid. As a plug-in to AutoCAD developed in VBA, the user could decide between two principal modes of analysis – (1) passive or (2) active – and three sub-sets of evaluating passive layouts: (a) inside and outside, (b) inside only and (c) outside only (four total types of analysis). 'Passive' refers to an all nodes to all nodes analysis, which Benedikt visualized as contoured isovist fields and Turner as a map of all neighbourhood size values. The author named this analysis mode 'visual integration' to indicate the visual exposure of each position node to all other nodes, rendering as a colour gradient the level of passive supervision at each position.

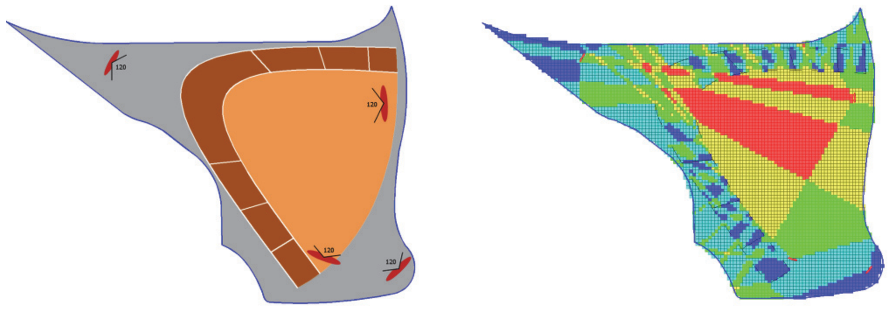


***Fig141**. OverView GUIs (2005) and 'active analysis' mode for 4 positions with 120° FOV and set directions of view, resulting (right) in a five colour gradient map for areas that are visible between those positions and FOVs*

The passive mode could be differentiated into three sub-sets of sampling from 1a – 1c, where 1a analyses all isovists across the site, integrating building interior with exterior site areas into one map. 1b and 1c essentially only analyse within different boundary polygons but the distinction is essential as the exterior wall is impenetrable apart from main entrances, not intervisible for varying light conditions (darkness outside, lights off inside but bright outside, reflections caused by sun position). The effects of separating analysis areas are striking (Fig141) and must be considered relevant to the analysis of thresholds between inside and outside. The separation also highlights the impact of the problem with the grid-based visibility graph: edge conditions or large open spaces will generate higher integration values, distorting the distribution in the middle ranges. Especially, for larger urban areas this is an issue where small to medium-sized spaces will be categorized as a single value band, when they might be very different.

The active modus allowed the user to select viewing positions by interactively selecting a node inside the GUI, specifying the FOV and setting the direction of view. Single or multiple viewing positions could be chosen to help analyse the constrained isovists for specific places of interest and their mutual exposure.

Another distinction was made for visibility across 'voids', such as atria, occurring in many typologies like schools or hotels. Not distinguished by Space Syntax, voids are intriguing spatial types as they allow for orientation and light but not for movement, and thus facilitate visual integration but not accessibility. This type of spatial unit is omitted by Bill Hillier, Benedikt and Turner.

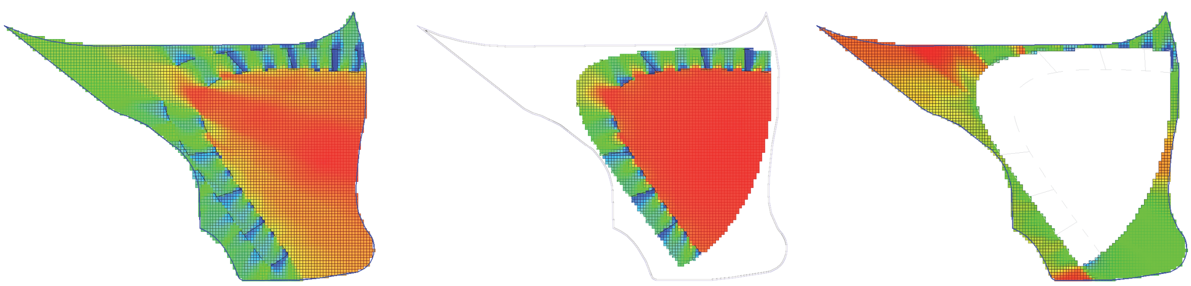


***Fig142**. OverView GUIs (2005) and 'active analysis' mode for 4 positions with 120° FOV and set directions of view, resulting (right) in a five colour gradient map for areas that are visible between those positions and FOVs*

OverView is one of the most successful developments of the first half of the 10 years of CDR and has been applied to many projects across all scales but mainly applied to building floors or masterplanning projects such as the 2006 Pristina Quendra masterplan in Kosovo in collaboration with 4M Group.

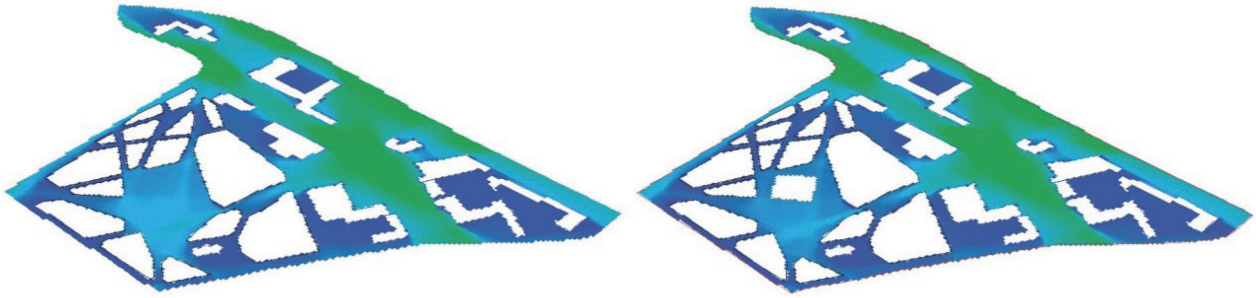


***Fig143**. OverView for Pristina Quendra masterplan in collaboration with 4M Group, 2005: (left) the proposed design shows the new central square to 'cast a shadow' of lower visibility where as a consequence (right) a building footprint was proposed for an additional public building*

OverView became integrated into the workflow at early conceptual design stages rather than at a design reporting stage where Space Syntax ltd generally operated, overcoming the 'analytical delay'. This delay in analysis often leads to compromises in the adaptation of insights as the design team will have considered new constraints and edited the design in the meantime. Integrating analysis seamlessly into the workflow enables designers to test and adopt insights without backtracking and its costs.

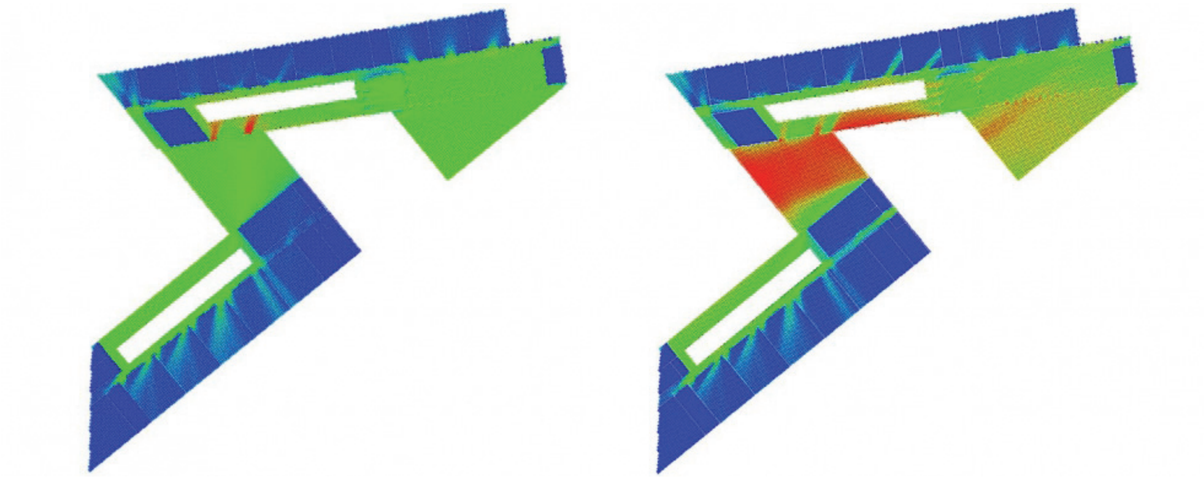


***Fig144**. OverView applied to the Darwen Aldridge Community Academy, Lancashire, showing changes in openings across 'voids' for increased visual integration*

Two more experimental prototypes were developed on the basis of OverView: Landscape and Multi-Floor. Landscape encoded a 2.5 dimensional ontology for the analysis of masterplans with strong height differentials, as it was always perceived a weakness in Space Syntax's VGA that topography would not affect visibility conditions. Multi-Floor was used for interior volumetric spaces to analyse the effect of vertical openings across floors such as atria.

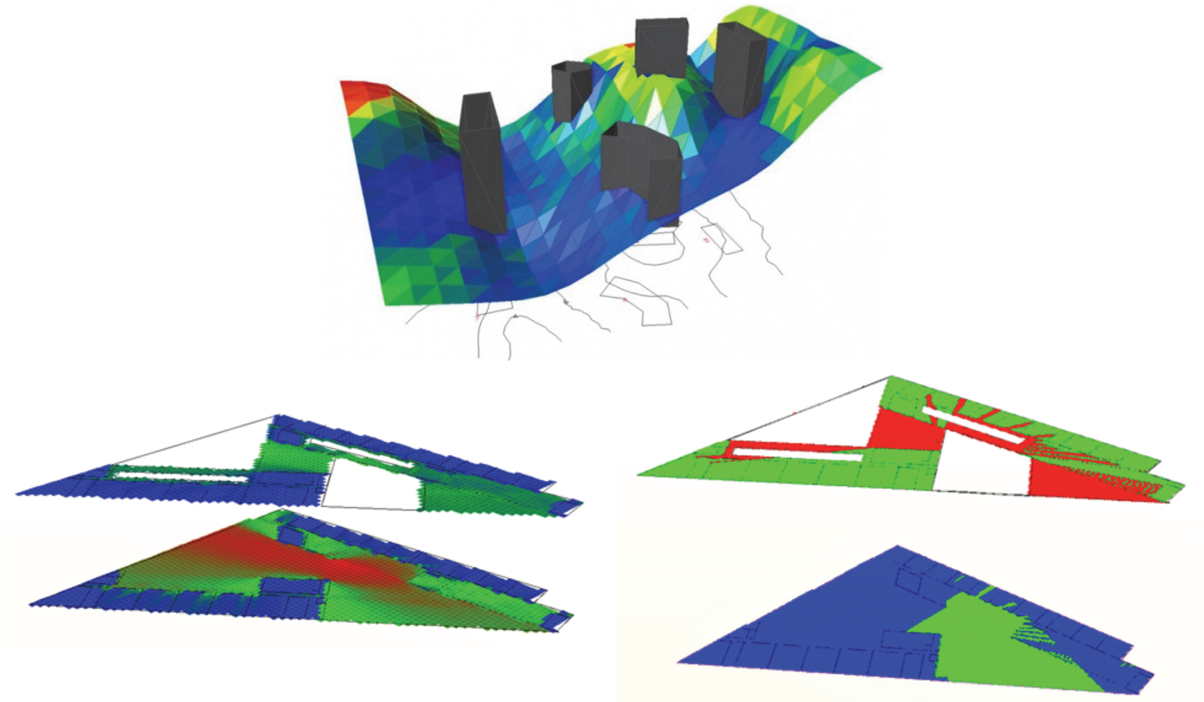


***Fig145**. OverView Prototypes, 2005: (above) on a hypothetical topography with buildings blocking views and (below) two modes of Multifloor showing (left) the 3D passive supervision of both floors and (right) an active supervision from the main atrium railing on the first floor*

OPTIMIZING THE ALGORITHM

After the SSSP project was completed, OverView was migrated to the new Java framework by Åsmund Izaki. The new Visibility Tool was developed in Eclipse using the Sunflow Rendering Library[47]. Sunflow partitions the target mesh into topological clusters of surfaces whose hierarchy is organized into a binary tree, called kd-tree (k-dimensional tree graph). A viewing position is sampled through ray-tracing by each surface cluster. Only cluster partitions where the ray is travelling through are eventually computing intersections (Fussell and Subramanian 1988). The intersecting surfaces are checked within the tree graph for proximity to location and direction. Only visible surface intersections are then rendered as visible. While the ray-tracing remains as in OverView, the pre-processed binary tree partitioning increases speed of computing, enabling real-time analysis and allowing the observer-user to sample positions and movement paths as series of positions intuitively.

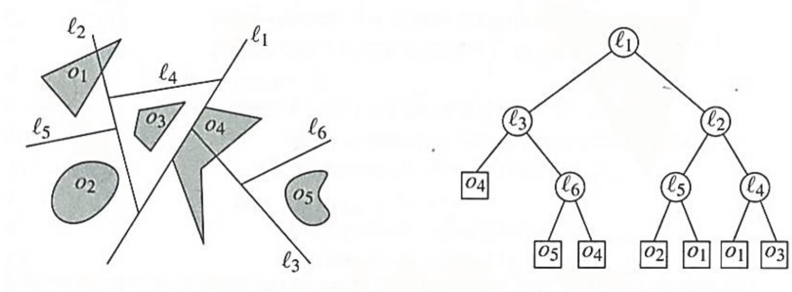


***Fig146**. Binary space partitioning tree (de Berg et al. 1997), showing objects in a plane (left) and their edge partitioning l1-l6 and the resulting hierarchical binary tree (right)*

[47] http://sunflow.sourceforge.net/, accessed 01.11.2014

Since 2008 the algorithm had not changed but the GUI and functionalities - including import and export file formats for workflow integration, number of analysis meshes for comparison and read-outs of resulting values - have been generalized and become a standard service within the practice and for consultancy. The tool was first applied to the Whitechapel CrossRail station Urban Impact study in 2009 as both a three-dimensional isovist field for the whole site and as four-dimensional visibility along access routes.

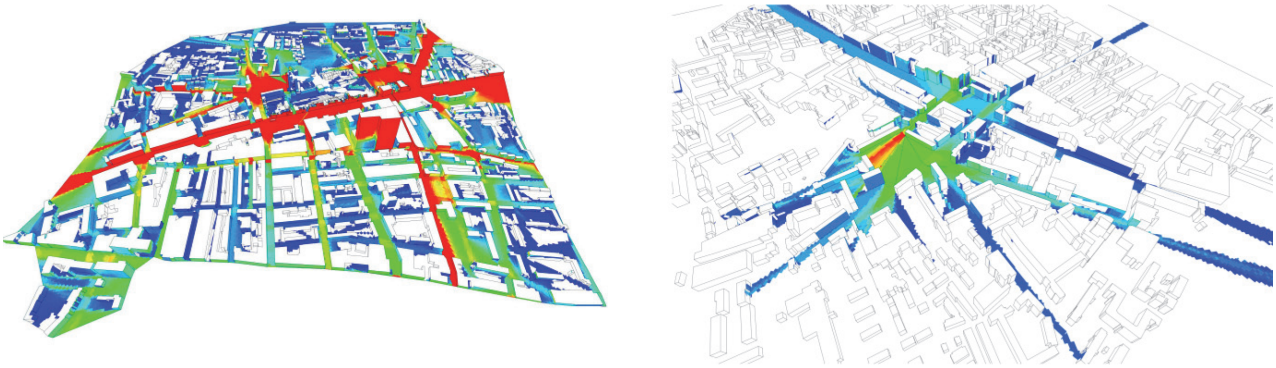


Fig147. *Visibility Tool on Whitechapel Station, 2009: visibility analysis for CrossRail Urban Impact study, showing (left) the whole site isovist field in 3d and (right) isovists along access routes to location of new entrances*

Particularly the dynamic three-dimensional isovist field (four-dimensional) produced heterogeneous implementations where local visual conditions of various types and scales of observers were approximated. The Whitechapel CrossRail station Urban Impact study supported the planning of crossings and visual connectivity by moving local commuters along sidewalks to the station entrance by adjusting the massing along elevations of Whitechapel road. Additionally, land-uses were identified according to visual exposure during different times of day.

A global pedestrian legibility analysis for out-patients via the four-dimensional analysis was conducted for the ongoing transformation of Guys and StThomas hospital in south London (commissioned by Tibbalds[48]). From observation, common routes for out-patients were identified from key departure locations such as London Bridge station, and visual properties of routes evaluated and forecast for 2012, 2015 and 2020, when all major developments were planned for conclusion (station, Shard and hospital).

48 http://tibbalds.co.uk, accessed 03.11.2014

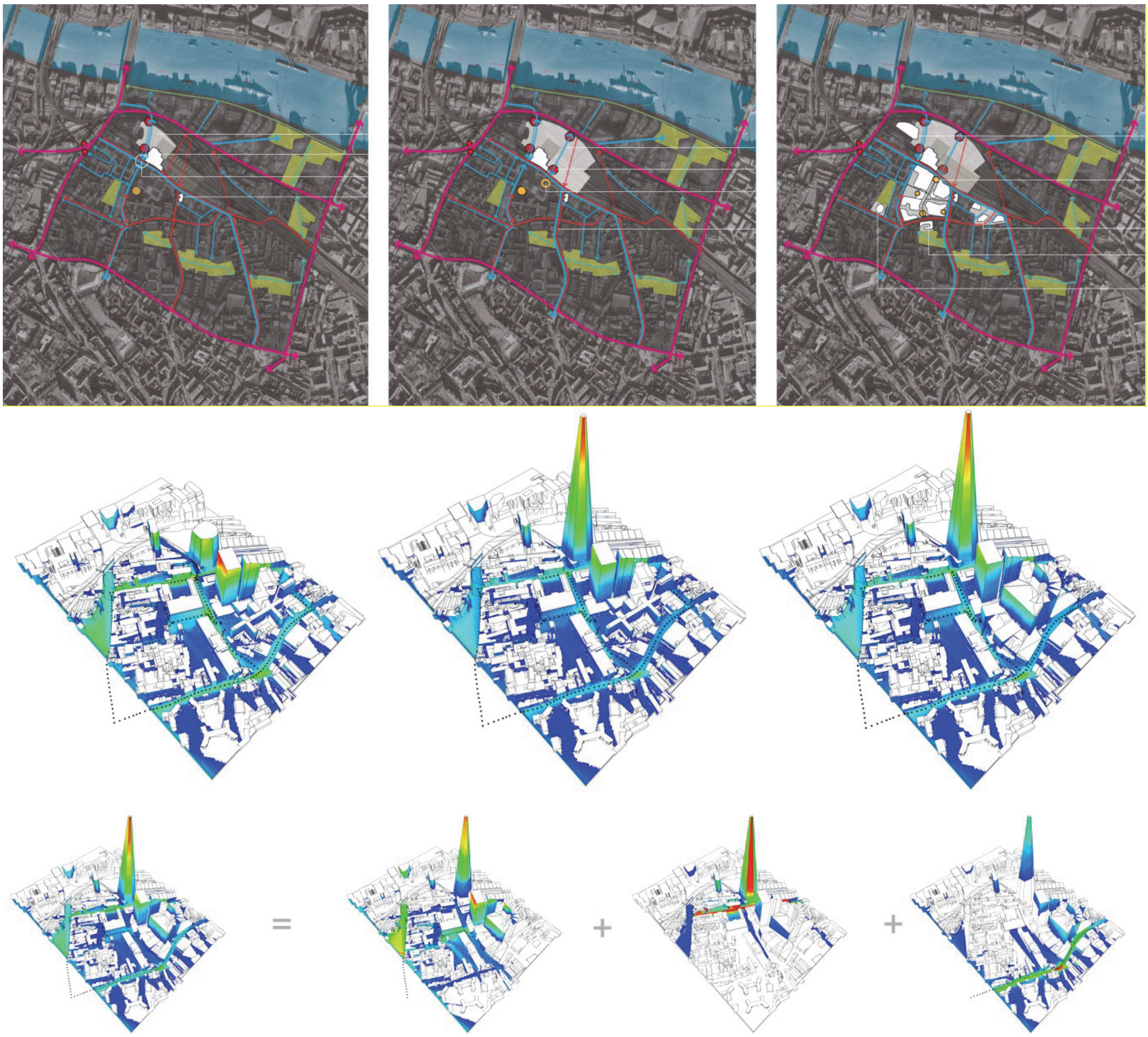


Fig148. *Visibility Tool on Guy's and StThomas hospital at London Bridge station, 2010: three access network scenarios for 2012 (left top and middle), 2015 (middle top and middle) and 2020 (right top and middle) identified by Tibbalds planners; bottom image, showing the compilation of a scenario through a series of access paths, where the exposure of the Shard can be seen to differ depending on the orientation to the analysed route*

Continuously, the scale of evaluated morphology and of the observer decreases, from an objectified sub-set of the public or a population of occupants to targeted decreasing groups of people where exposure relates to quasi-subjective performance. For the Euston Square station forecourt in London, commissioned by Transport for London (TfL), only local commuters were addressed from observed routes and only proposed building mass scenarios were analysed and compared rather than a neighbourhood. Eventually, visibility conditions can be reduced to individual windows or interior retail frontages, weighting the analysis to specific performances such as global viewer onto individual apartment, shoppers in movement or individual sales points and desks.

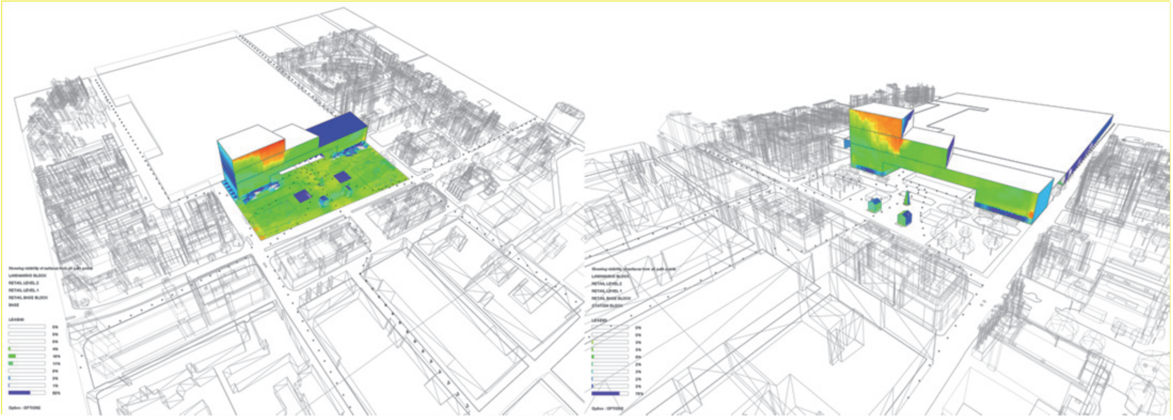


Fig149. *Visibility Tool for the development analysis of Euston Square, 2013*

The more specific the analysis becomes the higher the need for pre-simulation analysis such as conducted on Whitechapel Crossrail station, Guys and StThomas hospital or Euston station forecourt where data collected (by other consultants or CDR) on general pedestrian or selected groups of users.

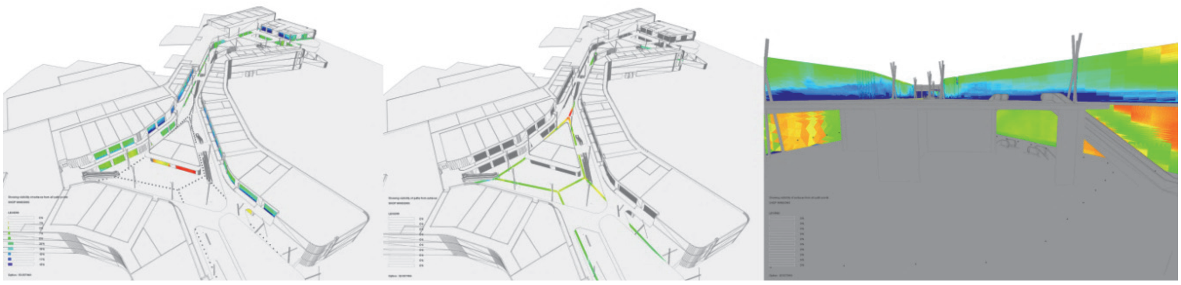


Fig150. *Visibility Tool for Packages Mall, Karachi, by Aedas architects, 2013, showing (left) individual shop front exposures, (middle) route segment exposure to frontages and (right) interior view of continuous shop elevations*

During the development phase for the European FP7 funded Resilient Infrastructure and Building Security project (RIBS)[49], Asmund Izaki of CDR re-developed Alasdair Turner's two-dimensional VGA algorithms to perform faster and provide additional measures of visibility (Izaki and Derix 2013). Izaki applied a visible polygon traversal algorithm (VPTA), doing away with the slower calculation of the visibility graph. While the standard visibility graph used by Turner (Turner et al. 2001) and CDR above (Derix et al 2008) evaluates every vertex on the visibility graph, polygon traversal only evaluates the visible vertices of a triangulated mesh, which has been generated from a building or urban plan, and does not conduct ray-tracing. A building or urban plan is triangulated into convex polygons and a connectivity list of polygons generated. The algorithm is a modified depth-first search (DFS) and traverses once counter-clockwise and then clock-wise from a viewing position. It starts from the polygon containing a view-point and first finds the nearest vertices of this first polygon within the direction of search. This specifies the edge of the next adjacency polygon from the list to traverse. If no two next vertices of the next adjacent polygon are visible from the view-point, then an open permeable edge is projected over the previous visible vertex, which intersects a visible edge of the adjacent polygon at some intersection point. Visible and open edges are compiled into a visibility polygon. The algorithm can handle plans with voids (Fig151) or complex non-convex perimeters (Fig152).

---

[49] http://www.cege.ucl.ac.uk/HIRG/Pages/RIBS.aspx, accessed 05.11.2014

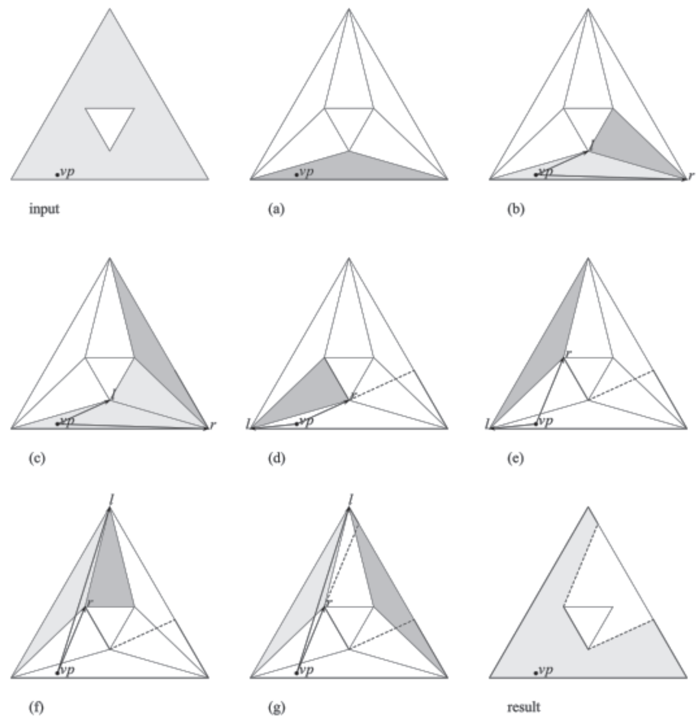


Fig151. *Visible Polygon Traversal Algorithm (Izaki and Derix 2013): showing both directions of DFS traversing adjacent convex partitions until the viewpoint cannot see the vertices of adjacent triangles; at which point an open permeable edge is created*

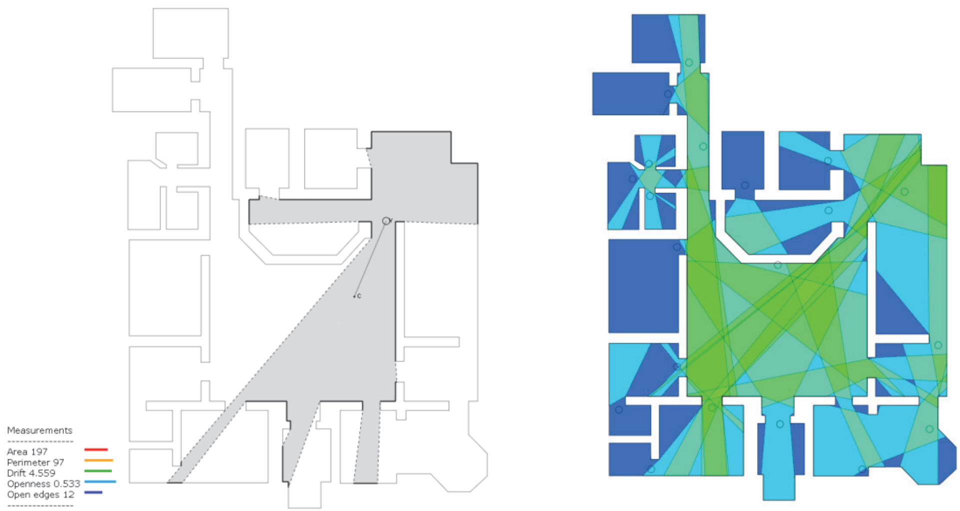


Fig152. *Visible Polygon Traversal Algorithm (Izaki and Derix 2013): (left) showing a single view-point isovist with permeable edges (dashed lines) and visible edges (solid lines) on Louis Sullivan's National Farmer's Bank of Owatonna; and (right) a multiple view-points evaluation of the plan with blue areas being seen only by one viewpoint and dark green being seen by multiple viewpoints*

Apart from being much faster and allowing for real-time interaction with the input mesh geometry, the VPTA makes multi-floor analysis easier because the list of adjacent polygons works topologically only, i.e. without a global hierarchy of dimensionality (like z-axis distinctions). In the context of the RIBS project, the fast multi-floor analysis provided the opportunity to evaluate the psychological effect of geometry on an intruder (here a potential terrorist) by being able to move the view-points about in real-time to find spaces where explosives could be hidden. Several measures were displayed to the user in the GUI to facilitate the identification of such visibility security risks such as

- Openness: measures the occlusion percentage of a view position by calculating the ratio between visible and hidden edges. The ratio indicates how porous the Isovist appears. The higher the Openness of view, the less hard visible edges are in sight and vice versa, indicating the amount of occlusion and therefore risk of concealment of explosives. Openness can also represent a positive measure such as 'Choice' of movement, meaning the amount of connected spaces a view position can 'see'
- Drift: measures the extents of a view, calculating the distance from the centre of a view to its extremities. Drift therefore indicates the distance decay mentioned above, utilized also by police to provide information on inhibition of an attacker or in the context of RIBS, the proximity to explosives
- Hidden Corners: the GUI rendered hidden corners on key command from which an observer could see the amount of risky corners behind which a threat could be stored

Additional measures are read out but are more standard such as area and perimeter. The measures were proposed already by Michael Benedikt (3.1) in his Isovist analysis paper (1979) but were not developed by him. Although the development context was specific to a certain typology (bank branches) and scenario (terrorist attack), the measures are valid for general situations of perceived risk within spatial environments (Fig153).

***Fig153**. New York Subway station: many hidden edges produce the sensation of threat as somebody could be hiding in the unseen areas adjacent to the view*

VISUAL SENSATION

The foundations for the Visible Polygon Traversal Algorithm (VPTA) were laid in a visibility analysis project for the National September 11th Memorial Museum (NSMM). In 2007 CDR was commissioned by DavisBrodyBond to develop the first three-dimensional visibility analysis for architecture, particularly interior spaces such as the museum (Derix et al. 2008). The task was to support the refinement of the geometry of a ramp that descends between the north and south towers of the former World Trade Centre. The ramp represents an emotional itinerary for the visitor who is led along certain impressions that build up the story of the event on September 11th 2001, a story of presence and absence. Hence, the visibility analysis was not to merely focus on simple evaluation of efficient layouts for supervision but to compile patterns of visitors' viewing propensities as a function of the volume of the museum. The measures introduced by VPTA aimed at behavioural pattern of occupants (intruder or generic user) were partially informed by the NSMM project.

***Fig154**. NSMM, 2007: (left) the bedrock space between the north tower and the slurry wall looking towards the ramp and the Last Column; (right) the ramp model folding downwards from the lobby (upper left) towards the slurry wall overview (right) and back down towards the south tower*

The key development for the NSMM represents the extension of the concept of the Isovist and VGA into three dimensions as the museum houses a rigorously volumetric arrangement and the ramp descending through the volume does not situate the visitor within a single plane of movement and vision. The objective of the analysis therefore was to evaluate the volumetric space, its visual properties and visual events emerging as the visitor travels down the ramp (Derix et al. 2008).

Three models of analysis were developed to evaluate and visualize the visibility conditions within 3D space and as a function of location sequences (4D):

a) Surface Visibility Analysis (presence)
b) Void Space Evaluation (absence)
c) Isovist Polyhedra (thresholds between presence and absence)

Each model aimed at accessing a different kind of affective response of the building geometry and the arrangement of artefacts (exhibits) on the visitor. The below three headings will shortly describe each model and their correlation to the visitor perception mapping. Models A – C were developed in the Microsoft Visual Studio IDE with the C# programming language, using the then new and barely tested API objects of McNeal 3D modelling CAD package Rhinoceros. Model D was developed in Java with the Java wrapper JOGL for OpenGL graphic and geometry libraries.

A - Surface Visibility Analysis (3D VGA)

The surface visibility analysis is a direct extension of the OverView Planar and OverView Landscape but works within an enclosed mesh. Like Turner's VGA within a bound planar polygon, the mesh is subdivided into a series of discrete positions in three dimensional space, here a 3D boundary representation surface. Two modes of evaluation are possible with this representation as discussed above: single active view creating a single instance of an Isovist in 3D or an all positions passive mapping of views creating what Benedikt called Isovist Fields, or Visual Integration. Like the OverView Landscape method, the visibility graph is produced by ray-tracing between all discrete positions in three dimensions connecting only positions that non-intersecting lines can link.

This method describes an exposure value of all building surfaces to all others, therefore indicating the probabilityof visual exposure to the visitor (Visual Integration). In the GUI the user could seed a series of positions to be evaluated as a sequence to emulate movement along a path. The methodology of surface visibility and exposure is representative of the notion of 'sensing presence' or remains, proposed as a design concept by the designer of the Word Trade Centre Memorial, Michael Arad.

B – Void Space Evaluation

For the NSMM concept a representation of visual exposure was missing that would describe the volume of space not as visible surfaces but as un-built void, since the museum space was meant to be a memorial to absence. Space had to be also measurable by its absence and artefacts evaluated by their partial appearance, through the curated itinerary across the void.

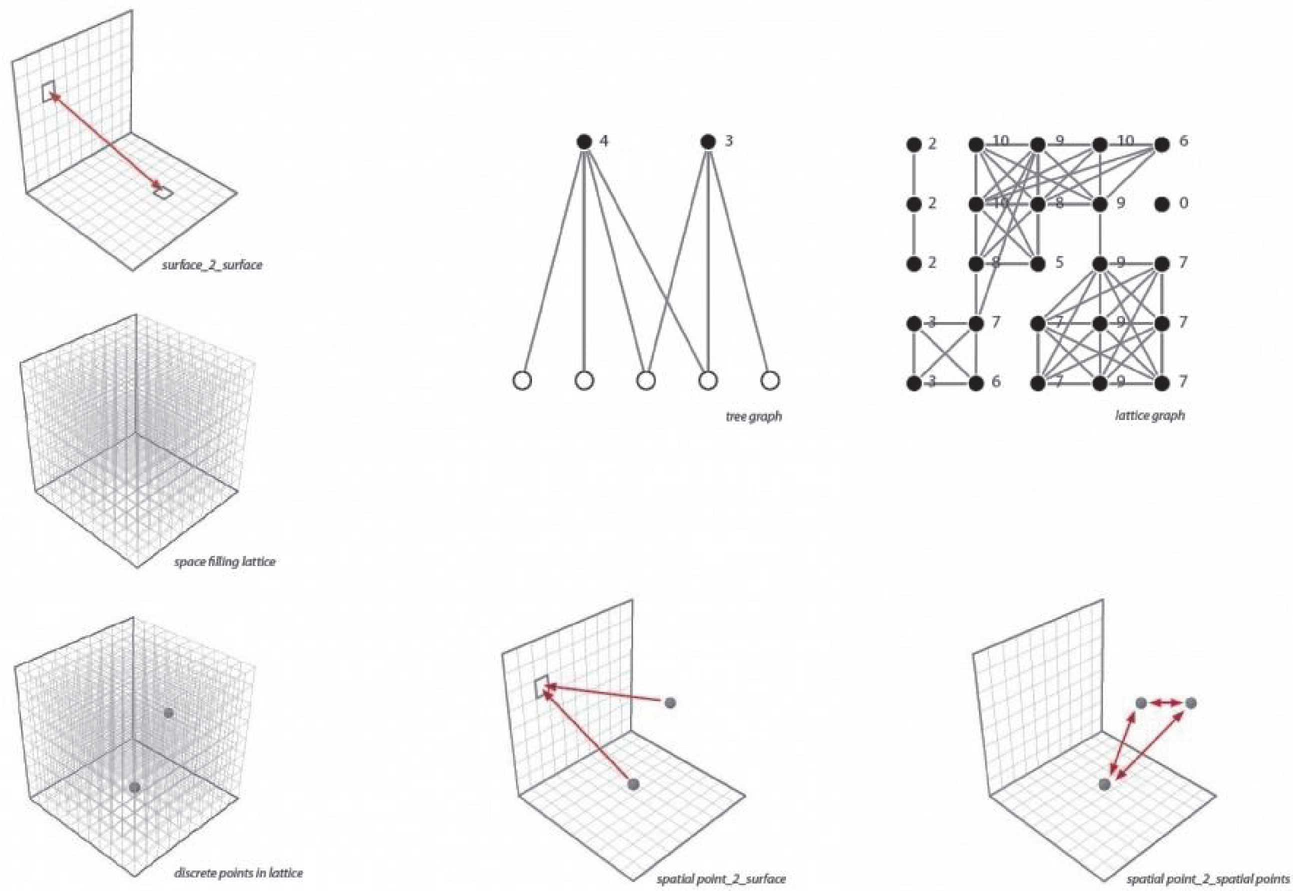


***Fig155**. NSMM, 2007: two discretization methods for volumetric visibility analysis: surface evaluation (top left) and voxel evaluation (bottom left) and hybrids (bottom middle and right), which result in either tree or network graphs (top right)*

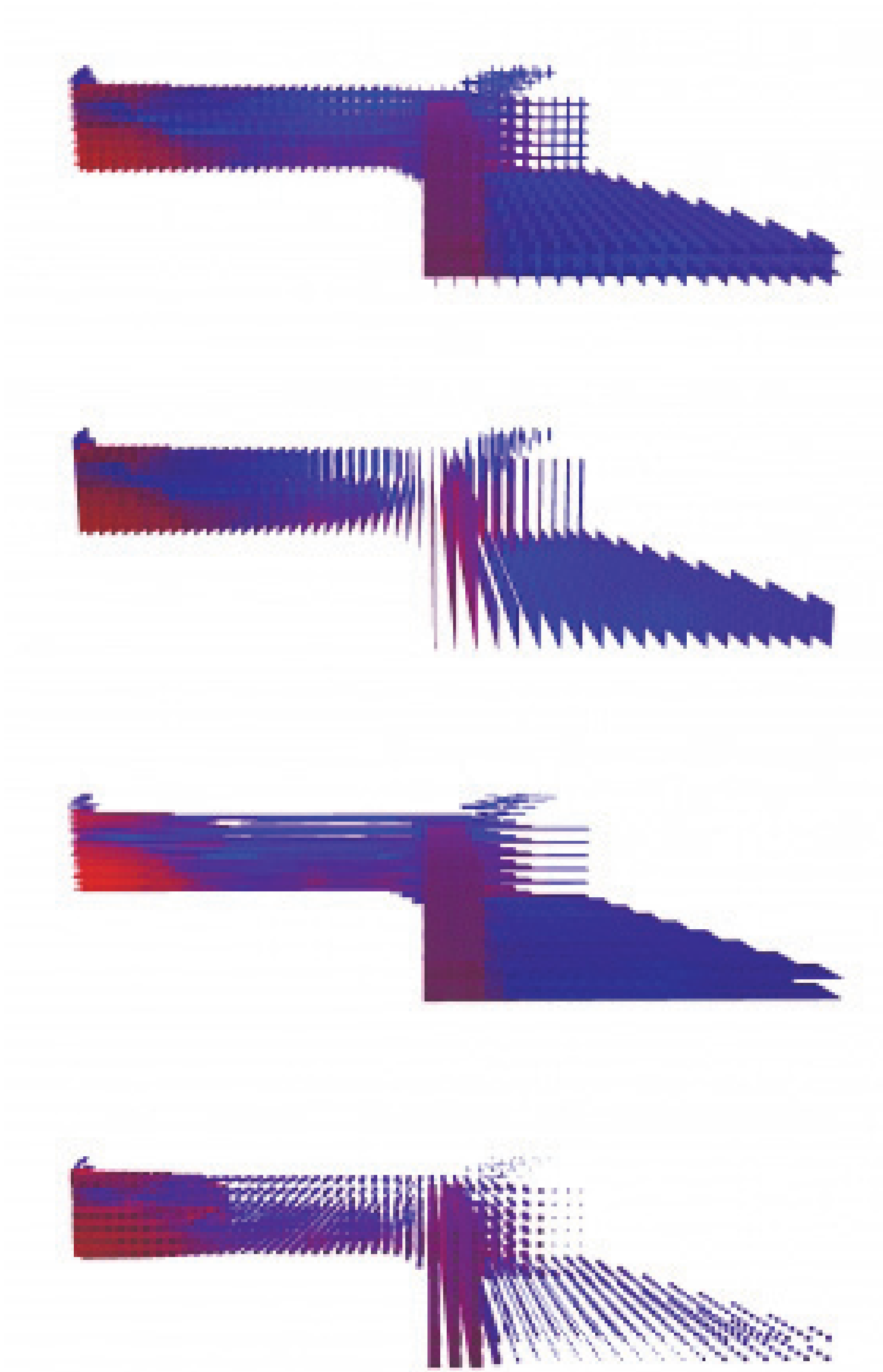


***Fig156**. NSMM, 2007: Void space visualization types showing (top to bottom) cross, vertical and horizontal sections, and a voxel representation of the visual integration properties of the interior volume*

The volumetric space of the bounding museum geometry was subdivided into voxels and Turner's VGA applied to this 3D lattice by ray-tracing, with both the active and passive visibility modes implemented. From the visual integration value of each voxel, visibility sections could be generated to allow for visual inspection of void locations where potential artefacts would protrude into the museum volume and give invisible value to sections of void.

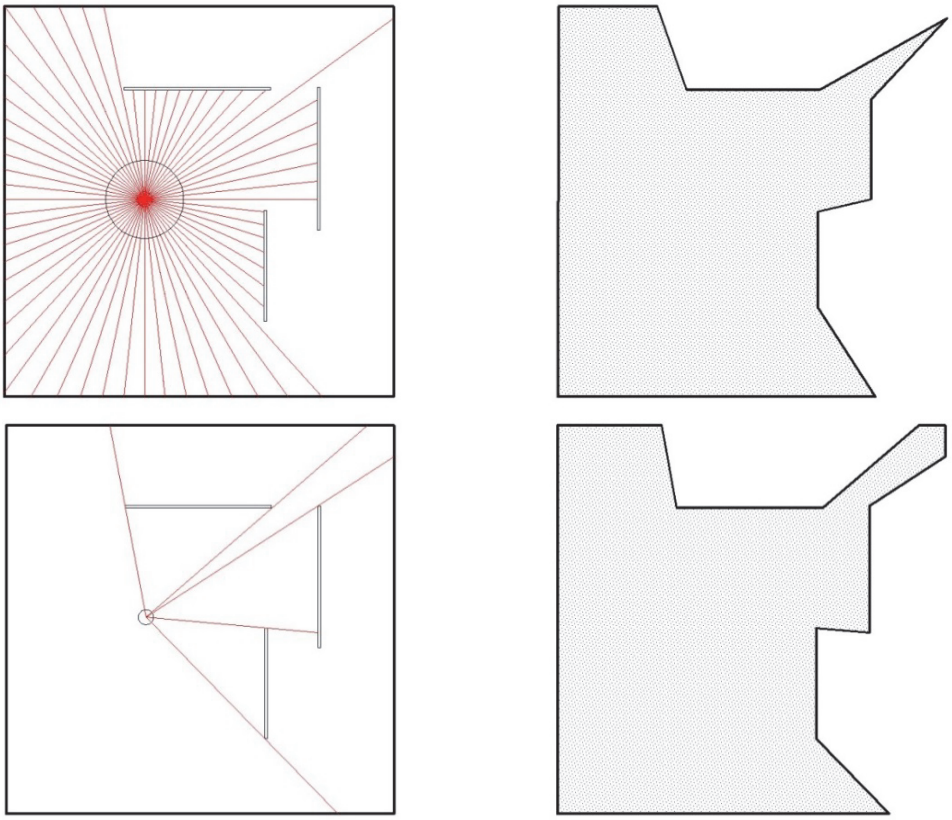


***Fig157**. NSMM, 2007: explanatory planar diagrams of the two construction methods used for the 3D isovist polyhedra; a simple radial-projection based on Benedikt's method (top) and an edge-projection method (Derix et al. 2008)*

C- Isovist Polyhedra

Michael Benedikt originally calculated the isovist via planar radial line intersections and linking the intersection points into a perimeter polygon for mathematical evaluation (Benedikt 1979). Benedikt's method was translated into three dimensional space producing polyhedra to which most of the isovist measures can be applied. The effect of a volumetric polyhedral viewshed on a viewing position however reveals expanse of spaces rather than planar access to visible edges. Properties of an isovist such as the directionality towards the centroid (gravitational centre) represent disparate behavioural probablities in two or three dimensions: in two dimensions the directionality points towards a location and a movement probability, whereas in three dimensions it points towards an attractor that directs the view, as the centroid might not lie in the movement plane. The polyhedral evaluation relates mainly to direction of the gaze rather than indicating movement as the ramp dictates the itinerary.

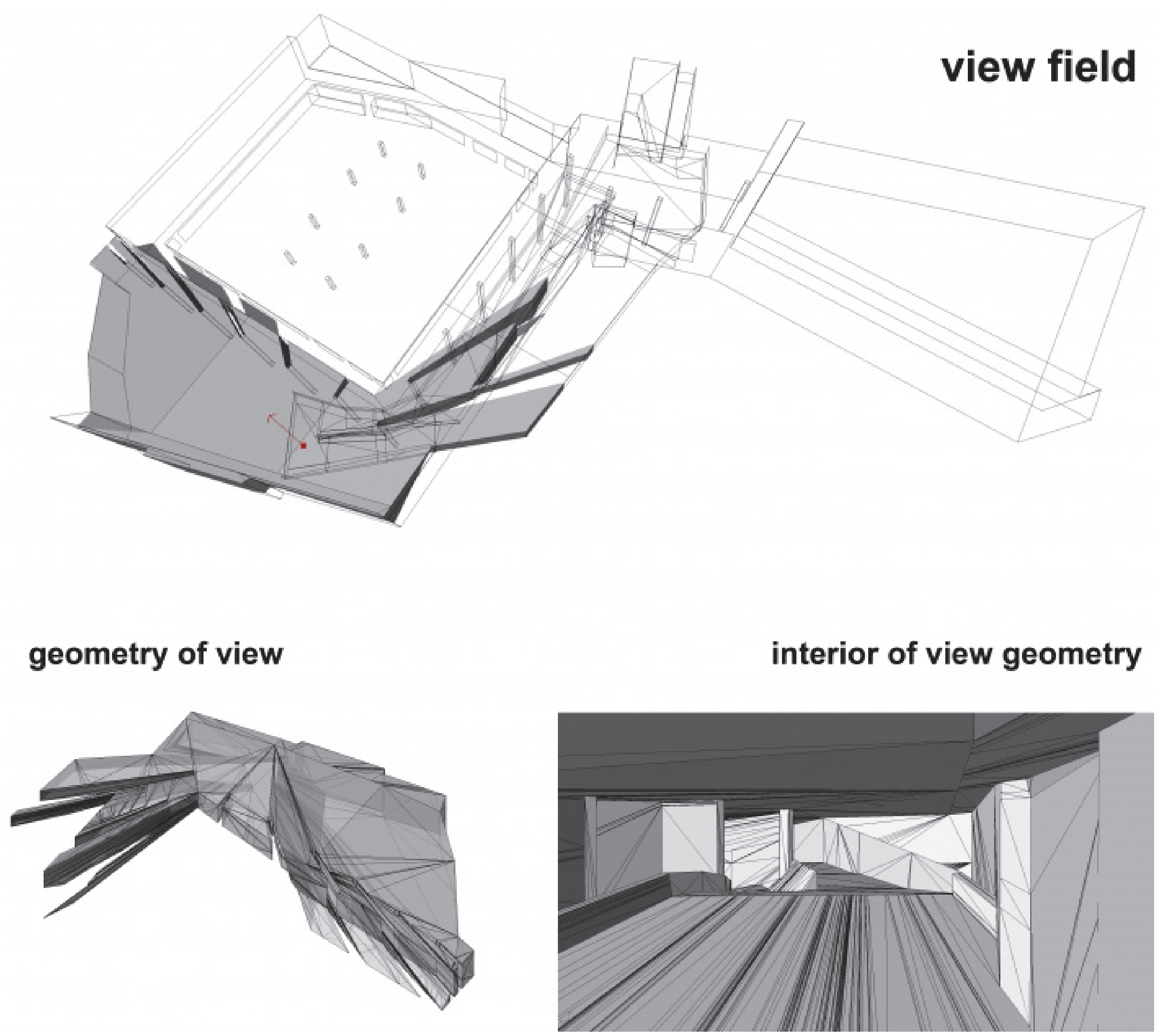


Fig158. *NSMM, 2007: a polyhedral isovist within the model of the NSMM and its gravitational vector from viewpoint to centroid of viewshed (top); the geometry of the polyhedral viewshed and an interior view of the viewshed (bottom)*

Two models of polyhedral construction were deployed: a radial projection and an edge projection. The radial projection represents a literal extension of Benedikt's isovist construction into three dimensions, whereas the edge projection generates a more precise geometry for evaluation. The radial projection projects the vertices of a sphere from the viewing position onto all visible surfaces and retaining the triangulation produces a polyhedron viewshed. The precision of the polyhedron depends on the resolution of the initial sphere and might not pick up all details of the building volume's geometry.

The edge projection method on the other hand projects all visible triangles (clipping polygons) of the surface mesh onto all other triangles of the mesh (subject polygons) as conducted in hidden surface removal using clipping polygons (Weiler and Atherton 1977). Where the projected and the subject triangle overlap, the subject triangle is clipped by subdividing it into smaller triangles that are subsequently included in the list of mesh triangles. The methodology is analogous to shining a light into a space and seeing edges projected onto deeper surfaces. When no more subject triangles are clipped, the visible triangles are converted into a resulting visible polyhedral mesh. This method produces a higher resolution and precise viewshed volume and computes faster, enabling real-time simulation on less complex spaces.

Walkable surfaces were subdivided into discrete locations and the isovist polyhedra generated from an eye-level height of 170 cm. The resulting polyhedra were evaluated for centroid, direction and size and the measures mapped into the museum volume, where the direction was visualized as a vector with magnitude for distance to centroid (the 'drift' property discussed in VPTA) and direction to indicate the orientation from viewing position towards the centroid.

Examples of Conducted Analysis

The key analysis aimed at understanding the generic visual properties of the museum space. Mainly methods A and B were used to reveal the overall hierarchies of spaces according to exposure and seclusion, as the two sensations were meant to be designed into distinct places in the museum for reflection or awareness of presence of others and surrounding space. Three highly integrated spaces were identified:

- double-height lobby space beneath the entrance hall (Fig159 A)
- space below the ramp turning point in front of the slurry wall (Fig159 B)
- central bedrock space near the Last Column (Fig159 C)

And one secluded space was highlighted along the ramp:

- the introductory exhibit space between lobby and ramp turning point (Fig159 D)

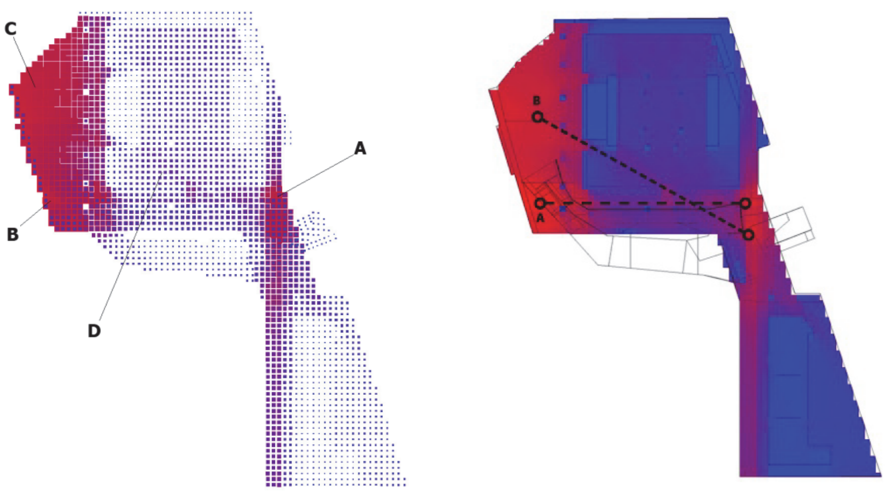


Fig159. *NSMM, 2007: Void Space Evaluation showing four locations of interest. A-C are highly exposed spaces while D represents an emerging visual link between two locations on the bedrock; (right) section at bedrock level showing two axes of visual connections emerge between locations in the same plane, generating potential movement*

Those salient spaces confirmed the design intentions that foresaw the double-height lobby space to be an arrival point at the end of the ramp between north and south tower (A); a place for reflecting 'absence'. Also the central bedrock space was chosen for the placement of the last column and hence anticipated for exposure (C). The space underneath the lobby and the ramp create a high exposure axis below the ramp that was not well anticipated (B) but supported the exhibition itinerary. A problem constituted a diagonal visual axis that emerged between the bedrock/ last column location and the movement axis below the ramp, because it would deviate visitors off the planned itinerary (D).

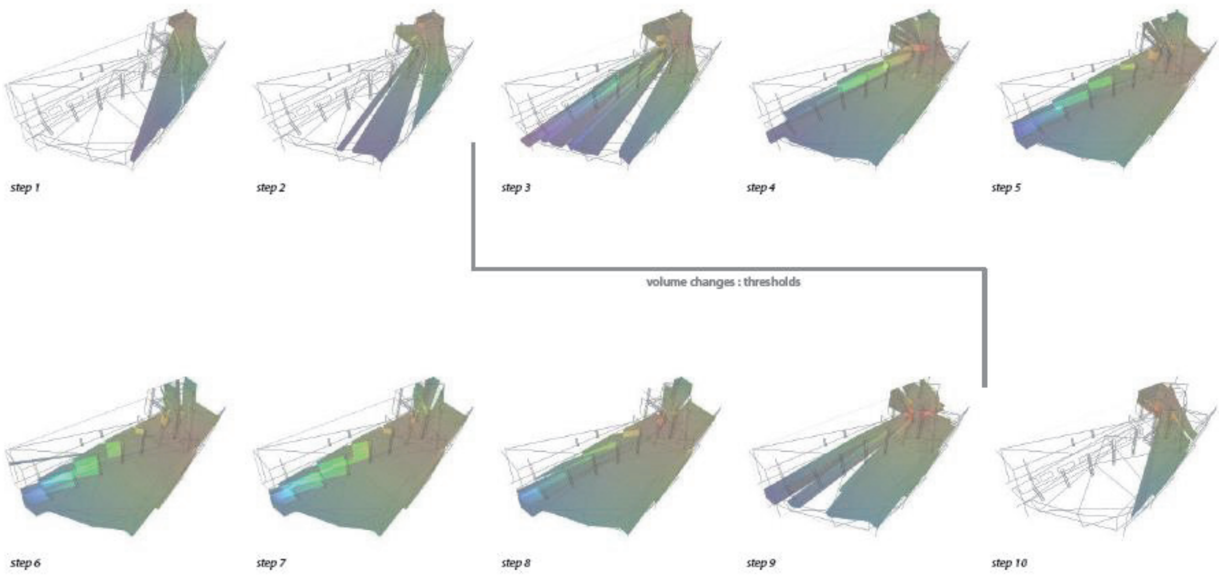


***Fig160**. NSMM, 2007: polyhedral isovist volumes along the slurry wall approach path on the ramp*

The ramp was analysed for change of enclosure/exposure, indicating the sense of extraverted or extroverted space, i.e. looking away from a location or focussing closely onto a viewer's present location. A series of path-based isovists through all three modes of analysis were produced at the approach to the slurry wall overview, resulting in a maps of change where the isovist size would jump and indicate thresholds of sensation from introverted directed views to extroverted un-directed views (Fig161).

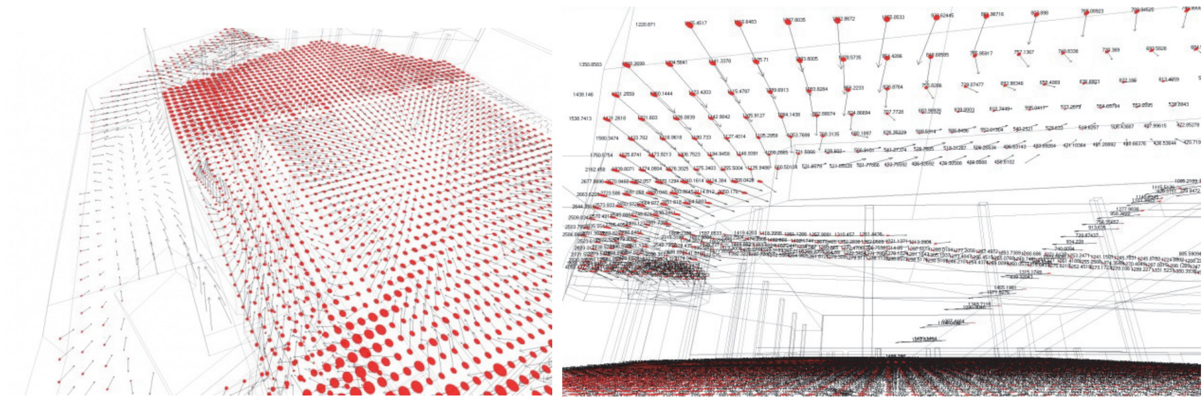


***Fig161**. NSMM, 2007: polyhedral isovist volumes directionality vectors mapped at their viewing position, showing the threshold of transition from intro- to extroverted space along the ramp, exiting towards the slurry wall approach*

This condition was more closely inspected via the directionality vector maps to search for an exact moment of change or transition. This threshold was found to be cutting through the introductory exhibition space along the ramp between the lobby and the slurry wall approach, meaning the visitors might be distracted out of the introductory exhibition towards the slurry wall. This meant that the geometry of the parapet enclosing the ramp at this stage along the itinerary had to be adjusted to re-align the threshold.

Viewer vs Voyeur

The examples of analysis show how visibility performance can be addressed without reverting to functional drivers like supervision or land-use allocation. The project report outlines a series of discursive measures that were proposed and partially applied

- Seclusion & Exposure
- Security
- Spatial Hierarchies
- Signage
- Observation & Direction
- Immersion & Scale (expanse & extension)

Those measures and developments were created in the context of exhibition design where the relationship between exhibits and viewers and space between both needs careful crafting to achieve the desired experience. The here developed three dimensional visibility analysis allows architects to carefully weight effects of such visual experience. The notion of framing views and exposure of a person represents the reverse of common visibility studies with an active intent of voyeur rather than being viewed. Clearly, such empirical sensations cannot be measured objectively due to subjective judgement from experience but the visualization of such conditions allows at least an informed design discussion about such situations.

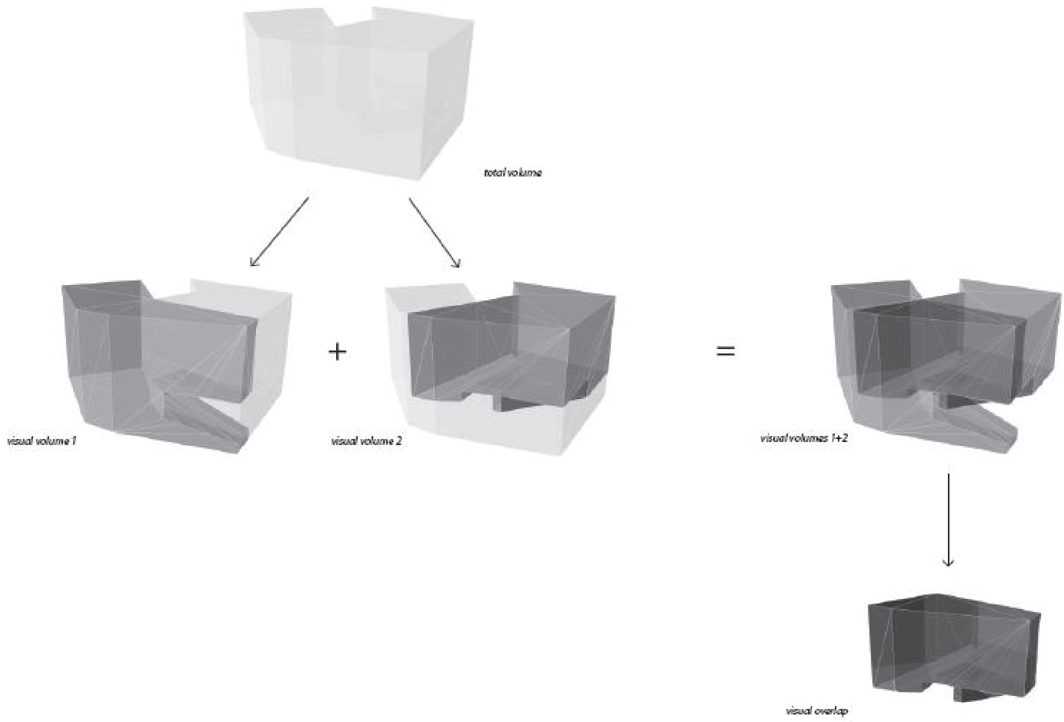


***Fig162**. NSMM, 2007: mutually visible space from non-intervisible vantage points generated from the polyhedral isovist; visitors' experiences of each other and sequences of space can be carefully crafted*

6.1.3 Summary: Mapping Measures of Perceptual Conditions

The field-bound perceptive maps introduced in this chapter are all based on discretizing the territory into a field of positions for which a value can be generated as a local measure. Those local measures most often relate the contextual geometry to the position and thus calculate some magnitude of condition that correlates to an occupant's perception like the understanding of size, scale, direction etc that in turn might suggest potential behaviours. The most common measures are based on either visibility or permeability conditions that often support each other. A series of correlations has been made evident that relate sets of algorithmic behaviours and measures to design objectives and KPIs based on perceptual conditions as shown in Fig163. From the table it is evident that behaviours and their measures can support multiple KPIs, and vice versa some design objectives can be composed of sets of behaviours. For example drivers behind pedestrian movement are almost all composed of algorithmic mapping methods because movement represents a generic problem. Most behaviours and measures inform this universal objective underlying nearly all architectural design problems. Visual Impact on the other hand is a more specialist objective and hence only correlates to few behaviours and measures found in design guidance (one might contest this of course and insist that the visual impact also underlies most generic architectural problems, yet the architectural community is rarely interested in its analytical performance possibly due to lack of regulation).

DESIGN OBJECTIVES / PERFORMANCES *correlate to* BEHAVIOURS / MEASURES	PERCEPTUAL CONDITION	Pedestrian access	Way-finding: local catchment	Way-finding: non-local catchments like tourists or out-patients	Crossing and entrance allocation	Land-use allocation	Signage	cycling and motorbikes	Footfall & flow	Destination loads (entrances/ exit dimensioning)	Frontage exposure (shops, buildings)	Sightlines / Desire lines	Passive supervision	Secure spaces	Space appropriation for safe places	Visual impact	Location value for apartments or shop frontage etc by visual exposure	Density & scale	Massing & Capacity
MOVEMENT																			
Obstacle Avoidance	proximity to objects	X		X	X				X										
Collision Avoidance	proximity to people + flow	X	X					X	X									X	
Seek	direction of objects	X			X	X			X	X		X							
Follow	heading of people	X		X	X	X	X	X	X	X								X	
Flock	alignment with people			X	X	X	X		X	X	X							X	X
Shortest Metric Local Access	directness/ time	X	X			X	X		X	X	X							X	X
Least Angular Local/ Global Access	ease / deviation	X	X	X	X	X		X											X
Topological Global Access	complexity / control	X		X		X	X		X	X								X	
Topographic Local/Global Access	effort / energy	X	X		X	X		X	X	X	X					X		X	
Hierarchical Access	expectation / order		X	X		X	X	X	X									X	X
VISIBILITY																			
Ray tracing	visible			X	X		X				X	X					X		
Optical Flow	change (image)	X	X															X	
Isovist/ Intervisibility	exposure	X	X		X	X	X		X		X	X	X	X					
Visual Integration	reach	X	X	X		X	X		X	X	X		X	X	X	X	X	X	
Visual Field Size	spaciousness			X		X					X	X		X	X	X	X		X
Openness	clarity (choice)	X	X		X		X		X						X			X	X
Occlusion	danger (safety)		X	X	X									X	X				
Drift	reach	X					X					X		X		X			
Directionality	pull			X				X	X			X			X				

***Fig163**. Objectives-to-Measures-to-Perception table of relations*

In a commercial setting there appears to be a tendency to apply two-dimensional discrete analysis maps for global performance of a whole building or whole site. In three dimensions – apart from the NSMM – discrete analysis is predominantly used to evaluate individual parts of a building or site. Also the four-dimensional analysis of the NSMM (path visibility) was mainly used for distinct parts of the building. This tendency might be linked to the resolution within each dimension where a two-dimensional map can represent a permeable field more robustly because the sample set is largely congruent with real possible states. Three-dimensional maps on the other hand, rarely capture true possible states and draw more heavily on assumptions.

6.2 GRAPHS | BEHAVIOURAL-BOUND DIAGRAMS

"Due to their minimalism and efficiency, graph-like mental representations of space are ecologically plausible, sufficient for the explanation of a wide range of behaviour, and, last but not least, they fit well to the neural structure of human brains"
(Franz and Wiener 2008, p577)

In the introduction to section 6.1, the distinction between maps and graphs was justified on two grounds: a) perceptual values vs behavioural affordance and b) territory vs diagrams. Maps discretize the field (abstraction of territory) that is to be quantified metrically and remain locally bound without abstracting the field into higher-order formalism. If the representation of a field needs to be instrumentalized for scenario planning, a higher-order formalism is required that abstracts the field beyond positions into ratios between elements in the field. Elements in a spatial planning context most often refer to geometric reductions of the built environment. They represent sets of aggregate positions that an observer intuitively generates from visual impressions or empirically from organizations, ordering elemental units into groups and groups into wholes like room boundaries, departmental groups or any other spatial typologies like circulation. Philip Steadman (1983) called those aggregate sets *dissections* of space and here they will generally be called (spatial) partitions.

Where in a discretized map the topological structure between positions is given by a subdivision method that generally results in an arbitrary partitioning of the field that approximates the territory as closely as possible. The topology between elements in a graph is either an interpretation of the graph or of some feature of the partitions such as doors as permeable links. A graph therefore is independent of the territory it represents and as formalism of edges and nodes is much leaner than a map. This allows for mathematical speculation on the geometric formalism, which can be used either for analytical or generative purposes. Michael Batty describes graphs as *"[...] the basic structures for representing forms* (i.e. partitions) *where topological relations are firmly embedded within Euclidean space."* (Batty 2004, p1).

Both, the partitioning from aggregates and their relating into topologies for interpretation reflects behaviours of spatial cognitive organization and of design heuristics. The aggregation function for the partitioning and graph generation process is mostly based on position values and therefore can encapsulate perceptual conditions. In fact, due to their abstract formalism, it could be argued that graphs are a vehicle to access and interpret perceptual qualities from geometric ratios such as the *laws of grouping* by the Gestalt theorists (Wertheimer 1938) like proximity, similarity, continuity, closure or symmetry proposed. Yet, graphs as higher order formalisms are not meant to be legible by lay persons but aim to provide the designer with a generative diagram. Hence, graph representations tend to be agencies for global observer perspectives.

Graphs were extensively used as a visual formalism to describe spatial structures by many architectural and urban theorists during the 1960s without necessarily applying interpretative functions for evaluating them. Lynch (1960) used graph-like diagrams to explain perceptual qualities of urban form using 'edges and nodes' as keys to draw mental maps which extracted topography from the territory; Alexander (1964) used graph representations to illustrate his mathematical theory of design. But graphs were predominantly used in geography since the first half of the 20th century to describe the geometry of geographic patterns and logistics such as Christaller networks (Haggett and Chorley 1969). In architectural computing, Steadman's layout representations through electrical flow graphs as a dual of standard plan graphs (March and Steadman 1971) provided an inspiration for space syntax's graph representations such as Hillier's justified graphs and axial line maps (Hillier and Hanson 1984).

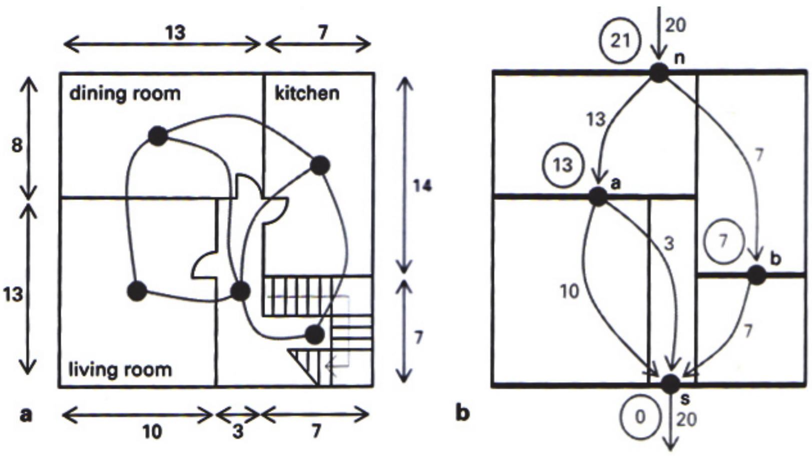


Fig164. *Steadman's electrical circuit graph representation , 1971: (left) a standard room layout or plan graph and (right) a dual partitioning for an electrical flow graph (taken and edited from (Kalay 2004, p257)*

This section begins by showing examples of simple partitions and a direct generation of a graph from a map. Then it introduces graphs for spatial analysis and eventually discusses an example of a graph as a generative design device.

6.2.1 Ordering and Relating

In order to build relationships between sets of spaces, elemental spaces need to be aggregated (or dissected according to Steadman) into groups with some definition. The organizing principle for a definition can represent various logics, for example Axial Line Maps relate convex spaces (Hillier and Hanson 1984, p91), layout graphs relate room polygons or permeability complexes relate accessible spaces. Those definitions work on the basis of some explicit conventional architectural aggregate already present in the plan, such as walls enclosing convex spaces, polygons for rooms and open doors for accessibility. Visibility maps discussed in the previous section on the other hand implicitly describe graphs such as the visibility graph. If the field is equally subdivided as done in VGA and OverView, the graph is just a reflection of the partitioning algorithm. But if the viewing positions are determined by some other definition such as workplaces within an office, visibility conditions and their graph generate partitions of higher order aggregates such as the distribution of workplaces by workflow organization or company hierarchy that are not part of explicit geometric conventions. In 2007 Pablo Miranda of CDR in collaboration with Henrik Markhede of KTH (Markhede and Miranda 2007) developed an isovist analysis tool that evaluated positions within workplace environments for their underlying graph-based performance. The prototype called SPOT allows the user to place or import viewing positions in a workplace layout, calculate the isovist field from each position and extract a graph that shows the network integration value for each location according to nested visibilities. The resulting graph shows the visibility network with edges indicating visible connections with line thickness visualizing distance and the nodes displaying the integration of a node into the network. The integration value here is called *relative asymmetry* and expresses the reachability of a node by all other nodes in terms of visual 'depth' (in how many isovist fields is a node embedded). SPOT is a simple example of the generation of a graph from the interpretation of a field into a higher order diagram, which can be used to investigate spatial performances.

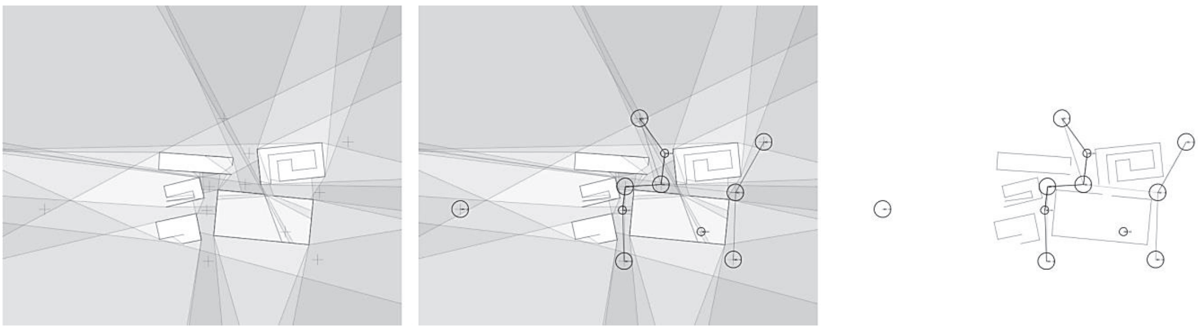


***Fig165**. SPOT, 2007: an abstract example of ten selected viewing positions producing (left) isovists, (middle) the graph from visibility partition and (right) the resulting graph only with node integration values*

6.2.2 <u>Movement Diagrams | from Maps to Graphs</u>

In section 6.1 the distinction was made between the visibility graph and the visibility mesh. The Fields model development illustrated how, despite the application of graph calculations like the Dijkstra algorithm, all positions on an arbitrary regular discretization were included in the mapping of the whole field. Therefore due to its direct analogy to the territory, the field could not be analysed beyond the values of local positions and consequently not instrumentalized. If on the other hand only a selection of nodes from the visibility graph is extracted subject to some definition, a graph can be built (called a sub-graph of the whole network of edges) representing a lean formalism from all ratios in the field. This independent formal abstraction no longer represents a map or mesh but an analytical and generative diagram.[50] Like discussed above, a graph is based on some partition or geometric reduction from which to calculate different performance states.

Another accessibility model was developed during the SSSP project (see 6.1) which was called the Routes tool. As opposed to the Fields tool, which calculated position values for distances to access points across the whole field on a mesh, Routes was developed to extract and visualize exact routes. Routes generates the visibility graph from the input geometry, namely vertices between closed polygons that represent walking obstructions, thus filling the permeable space with visibility edges. The Dijkstra shortest routes algorithm (see 3.4) generates the metric shortest paths between OD (origin-destination) pairs. Multiple destination points can be chosen and the algorithm selects the closest access point to create the graph. The edges of the graph are coloured as gradients of distance from red equals zero meters, to blue equals furthest distance. The edge weight is determined by the flow load, meaning that the input origin points can be weighted by number of people.

Fig166. *Visibility Mesh to Graph: (left) visibility graph as produced by VGA on a site in Athens (with students of IE University, 2013); (middle) visibility graph generated from geometry vertices only and (right) shortest routes graphs from same OD points shown in the middle visibility graph*

As described in 6.1 Reaching Agency for the Fields model, the Routes model also provides three analysis modes for route generation, using Dijkstra's algorithm to weight the visibility graph by metric distance, topological distance (least turns) and angular distance (least sum of angles).

Fig167. *Visibility Mesh to Graph: (left) flow field from the Field tool with four origin and two destination points; (middle left) metric shortest routes with same OD points; (middle right) the extracted graphs and (right) the two graphs weighted by least angular distance (which here are tree graphs)*

[50] The distinction needs to be made between a graph and a graph converted into a diagram, i.e. a graph consists only of nodes and edges reflecting topological and geometric relations. Graph drawing methods like the *force-directed layouts* of 5.2 represent a visual interpretation of the underlying graph into diagrams.

The Routes model has become one of the most applied design simulations of CDR and underwent many transformations. For the already introduced Whitechapel Urban Impact study commissioned in 2009 by the CrossRail consortium for the refurbishment of Whitechapel station in East London, the generated route graphs provided information on the number and location of entrances planned for the then current station design. The entrance locations and loads were evaluated for both performance and impact on context. It could be shown that the placement of two instead of three entrances was sufficient as one entrance was not very accessible and thus reduce cost of gates, staff and security. Alternatively, only one entrance would increase the walking distances beyond the desired access time of ten minutes. Additionally, depending on pedestrian user type (local commuter or non-local tourist), some streets and alleys where footfall was expected will probably not experience any pedestrian through-traffic (Fig168). Another study supported decision making on the placement of new formal crossings (as opposed to informal crossings where people cross without provision) due to shortcuts and visual access that they provide (Fig169).

Fig168. *Whitechapel CrosssRail station entrance configuration analysis, 2009: (left top) two entrances catching approximately 50% of commuters and showing that envisaged small alley behind station site might not generate footfall (b); (top right) least angle analysis providing insights for flows from hospital for out-patients and potential new crossing*

Fig169. *Whitechapel CrossRail station crossing scenarios, 2009: (left) one extra and two extra crossings showing that one extra crossing does not enhance the walking times or add extra load to the eastern entrance; (right) same scenarios with extra noise added to flows via perimeter origin points*

The two dimensional Routes model was extended for three dimensions for the 2009 mixed-use neighbourhood MIST 340 of Masdar Zero Carbon City project in Abu Dhabi. The residential blocks of the city contained vertical circulation in a high temperature climate for which the entrances in relation to public transport access points were under scrutiny to constrain walking times to less than two minutes. It was assumed that longer access times would adversely affect apartment rental values.

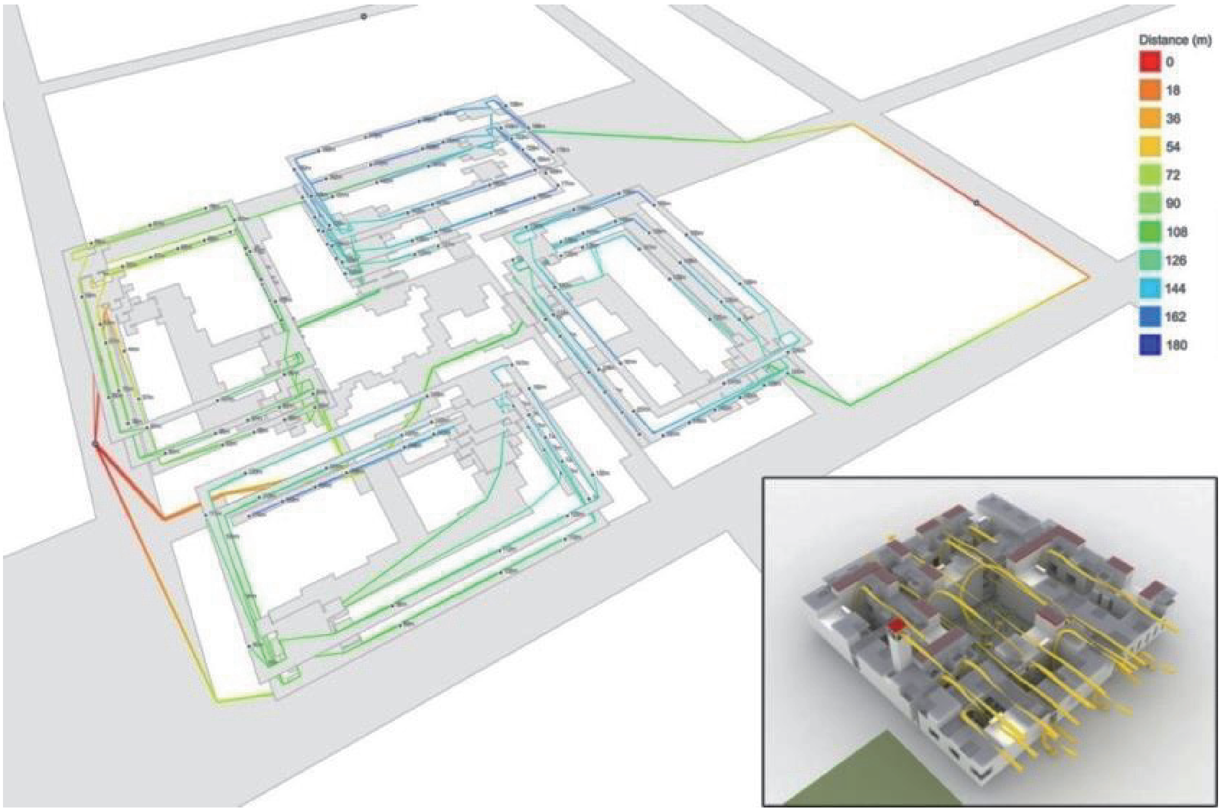


Fig170. *Routes model in three-dimensions for the MIST 340 neighbourhood in Masdar Zero Carbon City, 2009; (bottom right) the air-flow diagram showing where higher flows coincide with routes*

The three-dimensional Routes model automatically interpolates link nodes into the visibility graph between two floors at landing points, so that vertical circulation can be included in the measuring of distance and angular sums can be approximated. Since Masdar, the three dimensional Routes model has been revised to work faster on the *Visible Polygon Traversal Algorithm (VPTA -* see 6.1) developed by Asmund Izaki. Because the VPTA does not pre-process a visibility graph but calculates directly on resulting triangulation edges with DFS, the route graphs are generated much faster than calculating all edges via a BFS used in a Dijkstra algorithm on a visibility graph (Izaki and Derix 2013). Because there is no pre-processing, designers can manipulate the geometry directly in the interface and observe the routes adapted to the new layout in real-time.

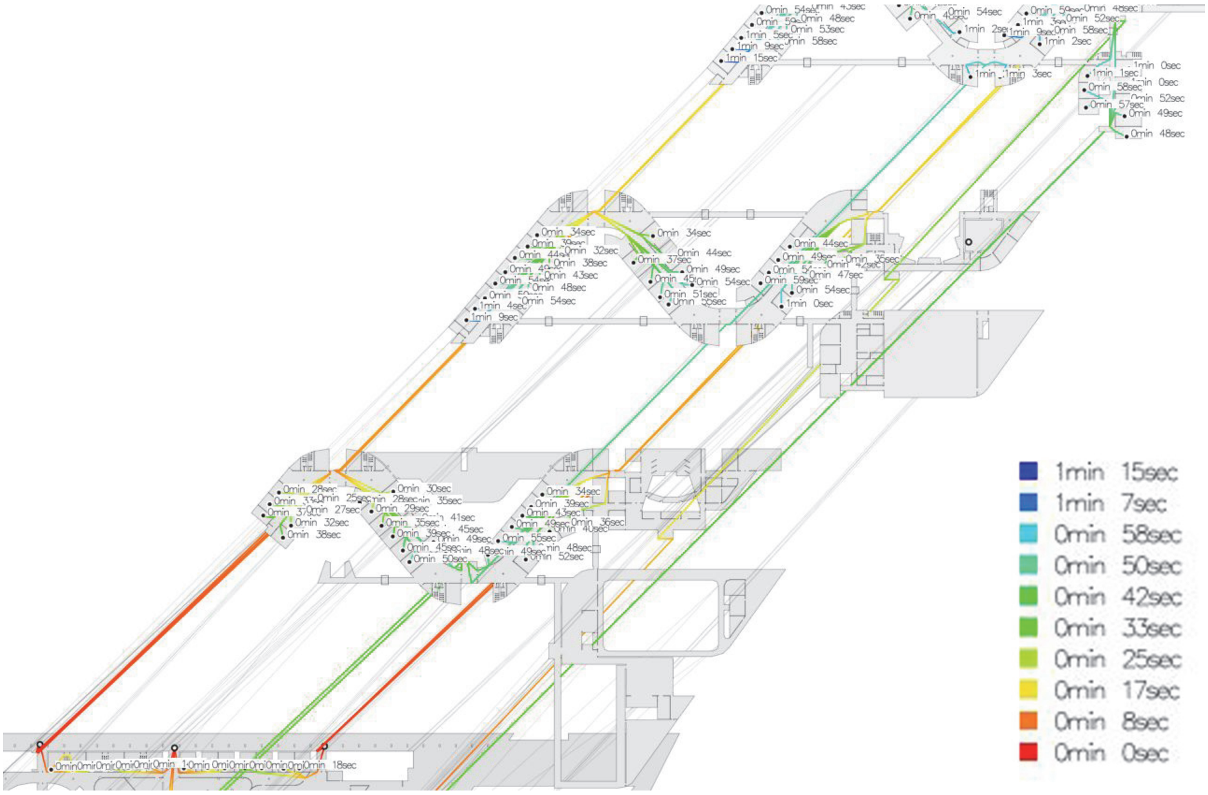


Fig171. *New Routes model using the VPTA, 2013: due to its faster calculation method, more complex building layouts including multiple floors can be evaluated for routes (here an exploded axonometric of the American Academy in Dubai, 2013); the larger dots at bottom of red lines are destination points where flows from departure points arrive that are located at higher floors*

Finally, a GUI component has been added that allows for pairwise OD node type association, avoiding only the nearest access node to be selected. Graph edges can now be bi-directional where flows along an edge can be summed up to reflect simultaneous opposing loads as conducted for the French School in Singapore (Fig172).

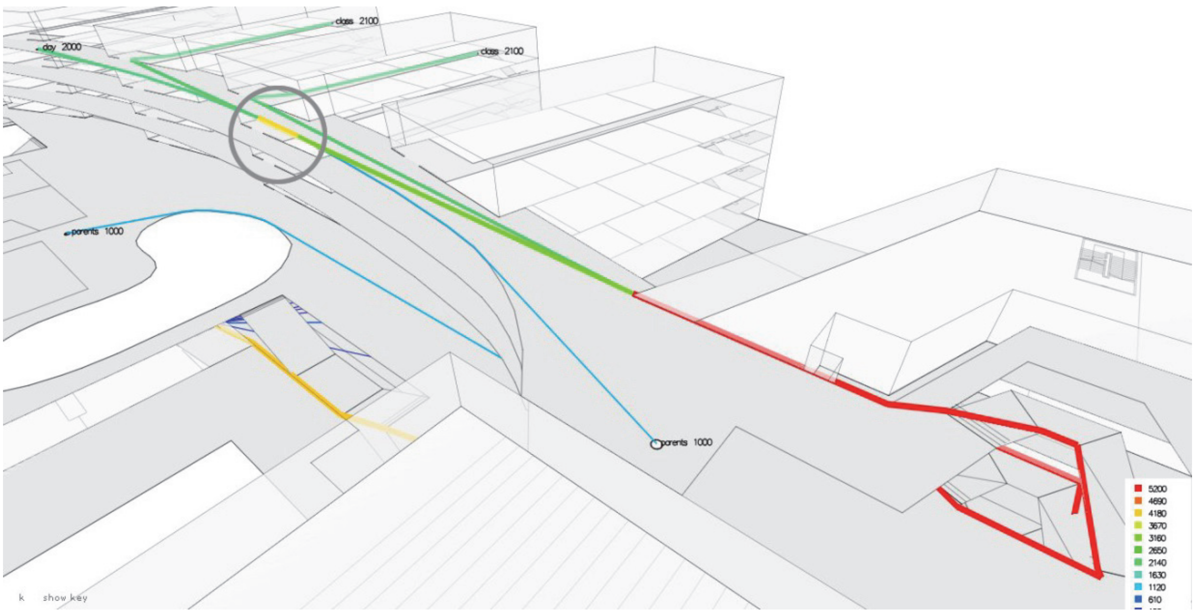


Fig172. *New Routes model with multiple simultaneous OD pairs on a bi-directional graph showing (in the grey circle) a summed up edge where two routes run through the same space and create a larger load (French School, Singapore, 2013)*

Summed bi-directional edge weights were also used in the Barclay Cycle Hire[51] journey visualization developed by CDR with Pablo Miranda in 2011. Barclay Bank was the sponsor of the London cycle hire scheme and in 2010 when the first million journeys had been completed over the first four months of usage. Barclay allowed the journey data to be freely used and CDR was commissioned to create a time-based journey visualization of the potential street network use.

***Fig173**. Barclay Cycle Hire time-based journey visualization, 2011: (left) the raw data from Barclay's, (middle) the centre-line interpretation of the OSM data and (right) the Barclay Bikes Cycle Hire docking stations in London's zone one in Google maps*

The visualization uses the OpenStreetMap[52] (OSM) road network of London to calculate the street centre lines into a network. As researched by Alasdair Turner, motorbikes and bicycles are more likely to use least angle routes (Turner 2009) to avoid deviation from straight line journeys and as a consequence angular distance analysis was applied directly to the centre line street network. OD pairs from the Barclay bikes raw tabled data was used containing origins - location + time of pick-up at docking station - and destinations - location + time of bike drop-off at which docking station - for each bike ID. Because journeys differ in duration, several routes might share edges over time. Those edges sum up the number of bi-directional journeys at any time unit, which is given by the tabled data in minutes when bikes were used. The resulting time-based bi-directional graphs are visualized by extruding the street edges used by the route graphs by the number of bikes used along those edges. Additionally, a colour gradient between blue (low) and red (high) was applied to the extrusion to make the visualization more legible.
The Barclay Cycle Hire journey visualization was not intended for use by a designer but as vehicle to explore city use. The time-based visualization showed dynamic patterns like morning, midday and evening journey loads and measured the carbon savings in CO_2 kg compared to other modes of transport. Resulting static graphs and maps will be shown below.

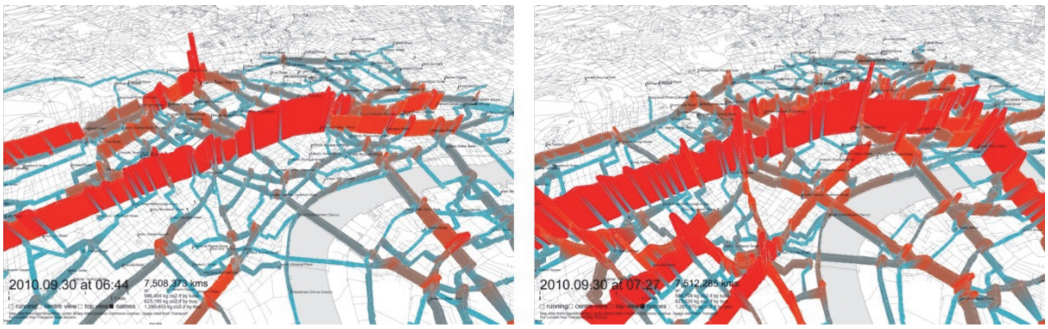


***Fig174**. Barclay Cycle Hire time-based journey visualization, 2011: two time frames of the visualization at 6:44 and 7:27 on September 30th, 2010; the higher an extrusion the more journeys are simultaneously flowing bi-directionally through an edge (street segment)*

[51] Now Santander Cycles: https://web.santandercycles.tfl.gov.uk, accessed 20.06.2015
[52] http://www.openstreetmap.org, accessed 07.11.2014

6.2.3 From Efficient To Redundant Behaviour

The issue with OD route graphs lies in the very efficiency for which they are used: to generate the least cost linkages (in terms of time, speed, orientation, energy etc) between two points. This segregates a site into catchments for specific heuristics of isolated user types. The last two models already attempt to allow for bi-directional edge weighting but eventually all resulting graphs are separate trees that do not allow for network properties such as cycles (loops in circulation) or choice of alternative routes depending on mixed heuristics or simply preferences.

The Smart Solutions for Spatial Planning (SSSP) project scoped six simulation models to be developed to create a digital chain for urban planning. The second stage model aimed at generating primary and secondary street networks. In isolation the two networks only represented tree graphs connecting primary attractors and contextual arteries for the primary circulation, and local points of interest for the secondary circulation. As aggregates the primary and secondary graphs produced a network that allows for different user behaviours and performances that go beyond the least cost route analysis.

The primary circulation was generated from a k-minimum spanning tree (k-MST), using the Dijkstra shortest path algorithm to calculate least metric distances between attractor nodes. The k-MST finds the tree graph that spans all or a sub-selection of *k* numbers of nodes from a whole graph, producing a least cost route tree as a reduction of the whole network (see O'Rourke (1994) for MST). This allows for the principal aim of urban planners that primary attractors like historic sites, squares or supermarkets should have high connectivity to engender activity along main arteries, which activate the neighbourhood.

Fig175*. Smart Solutions for Spatial Planning, 2008: (left) the primary street network tree graph generated from a k-MST; (middle) the primary network integrated into a generic grid of the total secondary network and (right) the resulting hybrid between primary and secondary networks*

The secondary circulation used the graph nodes of the primary circulation, a planner-defined grid mesh for minimum urban block sizes and secondary access points for local facilities (general practitioners, shops, schools etc). A network integrating the k-MST, new grid and locations had to be generated, which by its size would constitute a NP-complete problem and requires some meta-heuristic to find a solution within a 'meaningful' period of time for *designing*. The model uses a meta-heuristic of Christian Blum and Maria Blesa (2005) using an Ant-Colony Optimization algorithm (ACO). All edges of the MST-mesh network are weighted by pheromone values that the ants attach (at some evaporation rate) as they search for short routes between primary and secondary access points. At each time step, one of many k-MST is extracted that represents the least-weight spanning tree, where the cost equals the sum of all pheromone values. This is called an edge-weighted k-cardinality tree (Blum and Blesa 2005). Over time, as the ants learn the shortest routes across the site that connect all access nodes, the k-MST settles into a sub-graph of the overall network. Together with the primary circulation k-MST, the secondary k-MST-from-ACO forms a network with cycles and redundancy, allowing for choice in navigation and complying with urban design requirements for hierarchical urban structure planning (Derix et al. 2012).

Fig176. *Smart Solutions for Spatial Planning, 2008: (left) the ant-colony optimization algorithm (ACO) laying pheromone trails along the secondary network edges; (middle) four different states of the k-MST from ACO and (right) a final hybrid state of primary and secondary street network*

A graph can be analysed for topological properties relating to spatial features, which in turn correlate to cognitive performances of users. These measures are no longer conditions of the field but of the graph formalism and as Franz describes, *"permit the representation of inconsistencies and incomplete knowledge, factors that appear necessary to explain several empirical findings in human spatial cognition"* (Franz, Mallot and Wiener 2005, p2). In an architectural design context, CDR has been introducing graph theoretical measures on the SSSP project in 2008 and the MIST 340 neighbourhood for Masdar in 2009. On SSSP, one of the six urban planning phases comprised a prototype for a massing model that determined the street aspect ratio from the integration of street segments into the global access flows. *Betweenness* centrality (see 3.4) was used as a network measure to weight the street segments most likely to experience high through-traffic footfall. Through-movement in the SSSP case is determined on the basis of shortest routes between all access points of the public transport network surrounding the site and summing up the number of times a route passes through. As discussed in 5.2 *Reaching Agency*, PTALs and access levels strongly correlate to the scale and density levels of an area down to individual streets. Scales in turn determine street widths or vice versa by the street-elevation aspect ratio (CABE, 2000). Street hierarchies are generated by pedestrian access simulations such as described via the Fields tool. *Betweenness* centrality offers another hierarchical classification of streets based on integration rather than simply efficient access metrics. It is time and user independent and draws its validity from the geometry of the street network itself. It informs the street aspect ratio due to the maximum flows being expected along a street edge due to its hierarchical location. For the final classification of scale along a street segment, a coefficient between centrality and access footfall was established that hybridized the two measures of flow (Fig177). As a consequence, planners are expecting high activity along such street segments and can specify higher plot densities with greater street-elevation aspect ratios, as proposed in collaboration with LB Newham and LB Tower Hamlets regeneration planners for the SSSP massing model. While the graph theoretical approach seemed valuable, the notion of integrating too many measures and performances into a single massing model halted the development of this particular model.

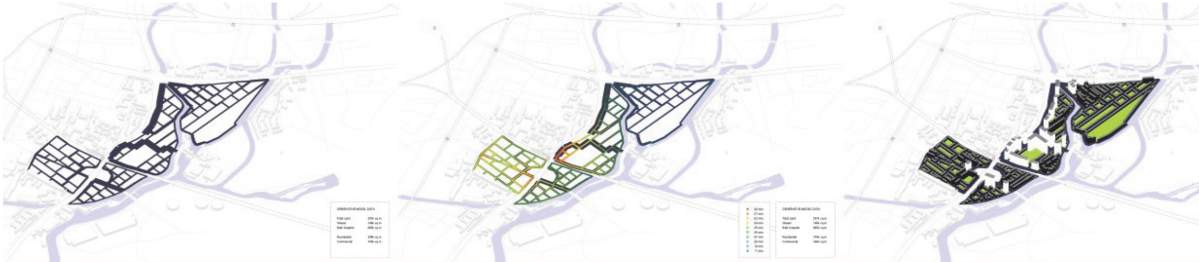


Fig177*. Smart Solutions for Spatial Planning, 2008: the first massing prototype showing (left) the edge betweenness centrality, (middle) overlaid with colour gradients of shortest distance access times and (right) the resulting height and scale interpretation using pre-established street section types*

Centrality measures were also proposed by Paolo Crucitti (Crucitti et al. 2005) to calculate street integration values such as those generated by the Axial Lines map. Instead of calculating a dual graph of the axial map from sightlines, they show how centrality measures - on what they call the *primal* graph - arrive at similar results yet allowing for better measurements of graphs into distinctions between street and place. Crucitti's method was used on the Barclay Cycle Hire visualization using the street centre lines derived from OSM as an undirected graph to calculate the *betweenness* centrality of the street network of London's zone one, with the docking station locations serving as nodes for OD pairs. OD pairs for through-journeys in this case were not calculated on the basis of Dijkstra's shortest paths but on the least-angle sum as proposed by Turner for cycling behaviour (Turner 2009). Only pairs that were actuated within the Barclay Cycle Hire journey data table were used. The application of angular *betweenness* rather than another type of centrality such as *closeness,* seemed reasonable for the Barclay Cycle Hire scheme because cycle journeys represent through flows across the centre rather than simple topological integration like *closeness* centrality (see 3.4).

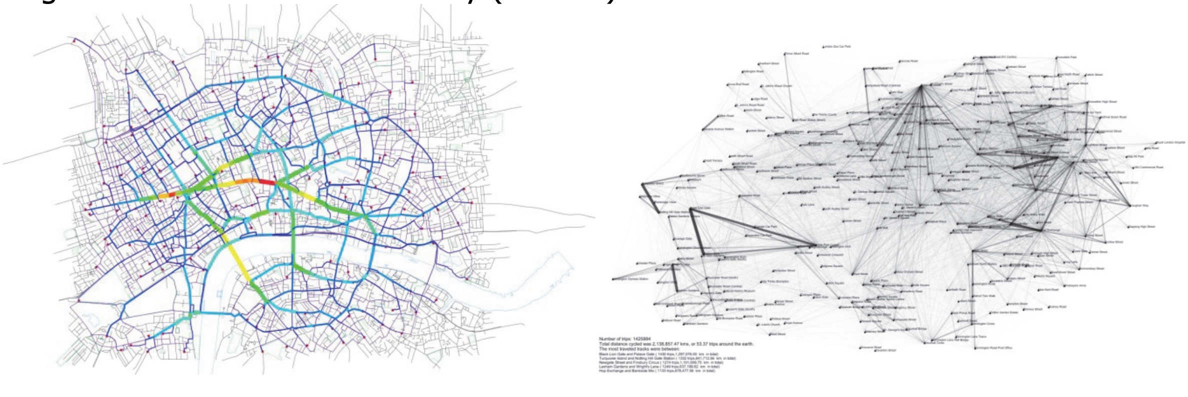


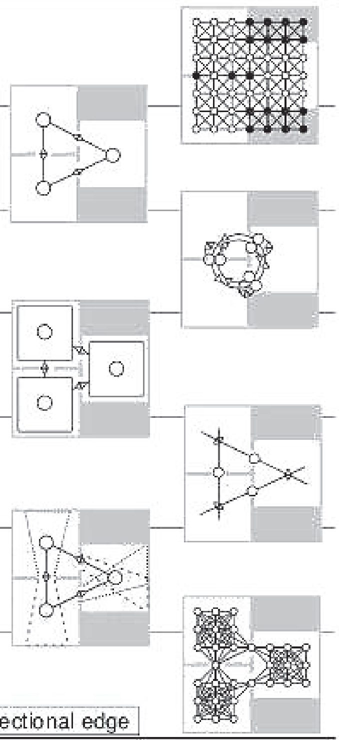
Fig178*. Barclay Cycle Hire network analysis, 2011: (left) colour gradients of betweenness centrality levels on each edge of the network graph, with blue = low to red = high, as a summary of all the journeys; (right) connectivity map by docking stations use*

The time-based visualization was supported by the graph theoretical analysis of *betweenness* centrality and a docking station connectivity map, showing which locations were well connected. As opposed to the time-based visualization, which reveals dynamic patterns across a day, month or season, the centrality map shows the street segments with assumed highest probabilities of flow, independent of time and the connectivity map attractions between locations.

6.2.4 Generic Network Behaviour

As mentioned in the 6.2 *Ordering and Relating,* all graphs require some form of partitioning that represent a higher-order aggregation of spatial information. Most graphs are manually generated or derive from an abstract second-order plan partitioning specific to a cognitive or behavioural quality as with visibility graphs. A consistent algorithmic approach was sought to generate a topological graph or network intrinsic to a spatial configuration derivable from the most elemental polygonal representation of a plan layout. Particularly in practice, a method is required that allows for a generic and direct network diagram generation, which holds the most elementary spatial information to analyse complex building or urban structures. Additionally, as shown in previous projects like the SSSP primary and secondary circulation structures, a non-efficiency driven representation of a network was required that enables many different modes of behavioural analysis including redundant and complex network elements such as cycles.

graph model	nodes	edges	pictogram
occupancy grids	xy-intervalls with obstacle probability	predefined, non-directional	
place graph	local position information at places	local navigation rules	
view graph	local position information at place transitions	local navigation rules, directional	
access graph	spaces	connectivity	
axial map	lines of sight	intersections	
isovist field	viewshed polygon	mutual visibility	
visibility graph	xy-intervalls	mutual visibility	



○ = node ◁ = directional edge ◁▷ = non-directional edge

***Fig179**. types of spatial graphs and their construction components (Franz el al. 2005)*

Rudolf Arnheim (1974) understood that a shape is composed of two elements: its boundaries produced actively by a creator and its *structural skeleton* perceived by the observer. In his 1954 book Art and Visual Perception – a Psychology of the Creative Eye, he proposed the *structural skeleton* to be like a *framework of axes* that provides the simplest obtainable structure from a shape, which determines the character of a visual object (Arnheim 1974, p93). He declared that this skeleton provides a shape with *characteristic correspondence*, giving a shape a typology that does not vary under (reasonable) transformation because expressions of a shape are associated perceptually to its character not its appearance. Inherently, Arnheim proposed a shape classification based on an intrinsic topological structure that is coherent with what is called a *medial axis* representation (see 3.4).

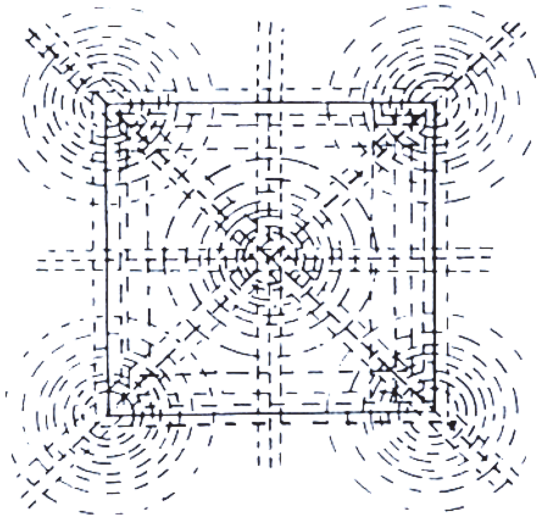
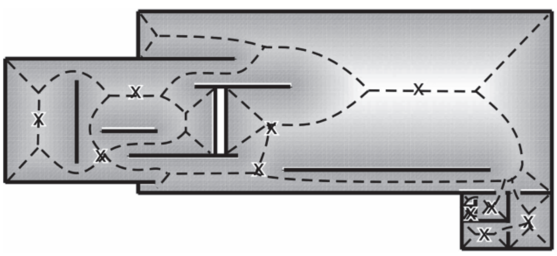


***Fig180**. sketch approximation of a Structural skeleton of a square by Rudolf Arnheim (1974, p13)*

There are many ways of constructing the medial axes of a shape as described in chapter 3.4 (distance field, tangent circles, straight skeleton, Voronoi construction). All of the methods essentially work on the principle of distances, radii or ratios relating proximities between two or more edges. Those edges need not belong to the same polygon and can represent different architectural elements, most obviously internal partition walls or elevations in cities. Arnheim linked structural *skeletons* to actions in space such as the painter or sculptor's hand movement in space intuitively negotiating the boundary and structure, while the observer perceives the skeletons by following with his eyes (Arnheim 1974, pp92-95). This behaviour-based perception is also argued by Franz and Wiener who relate medial axes to the spatial cognition via egocentric distances: "*Additionally proxemics suggests that humans evaluate other humans, objects, or even open space differently, depending on their egocentric distances. A close wall has a different impact on behavior than a more distant one. Therefore, the detected centers might be interpreted as positions that bring the surrounding walls in an experiential equilibrium. In terms of Lewin's field theory, these positions of balanced perceptual forces therefore also gain a special valence within an environment*" (Franz and Wiener 2008, p588).



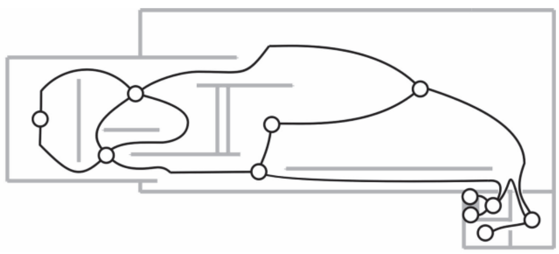


***Fig181**. Franz and Wiener's (2008) saddle lines from a wall-distance field corresponding to people's perceived points of spatial importance, which is regenerated via a skeleton to approximate a place graph*

Further, van Tonder used medial axes (constructed from the largest tangent circles algorithm, which he called *hybrid symmetry transform* (van Tonder and Ejima 2002)) to reveal locations within Japanese garden complexes at which shapes could be perceived in desired configurations (van Tonder 2004). 'Forks' within medial axes represent privileged viewing positions from which the topological structure of a spatial configuration is best legible. Again Franz and Wiener enforce this point later when stating that *"in the context of spatial cognition, the forks between nodes in the saddle line skeleton might also be of analytical interest. As is apparent […], these forks often represent spatial situations at which navigators have to draw decisions about their further path. In navigation experiments such decision points have been shown to have a special meaning"* (Franz and Wiener 2008, p589).

CDR used two models of topological skeletons to represent spatial polygonal structures: a) a straight skeleton-based prototype for urban analysis (Leymarie et al. 2008) and b) a discreet Voronoi based construction model for building analysis (Deleuran and Derix 2013). The 2008 model developed by the author and Pablo Miranda used Aichholzer and Aurenhammmer's *straight skeleton* algorithm (1996), approximating the medial axes with straight line segments. The *straight skeleton* uses a similar logic as the original Harry Blum *wavefront* algorithm (Blum 1967) by shrinking polygons to generate the vertices offsets along angular bisectors. The novelty of this prototype, which was called the Medial Axis Generator, was that it repeated the straight skeleton on the outside of polygons, meaning that skeletal ridges were generated from edges belonging to multiple polygons like wavefronts that can be applied without finding adjacent topological edges (as required for angular bisectors). This allowed the skeletons to be representative of urban public spaces as well as building interiors. The offset of edges from the shrinking process counts for the distance away from edges and can be visualized as a spatial proximity property. This is represented as a *height field* where the height of the skeletal ridges corresponds to the distances to the composing edges (Fig182).

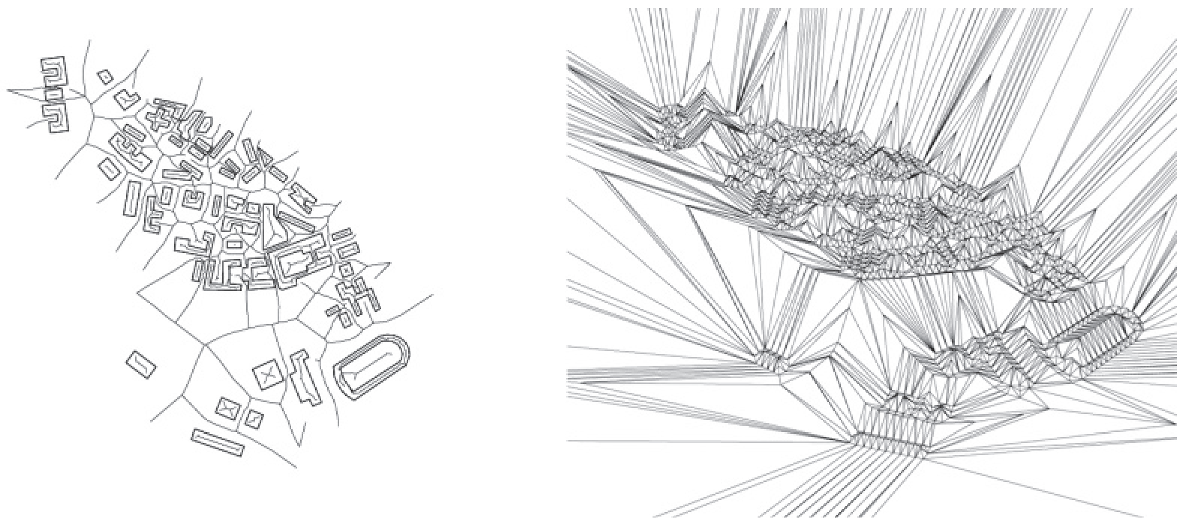


Fig182: *Straight skeleton of the KTH campus, Stockholm (left) and the height field (right) (Leymarie et al 2008)*

Three spatial measures correlating to perceptual qualities related to the resulting visualizations in the context of urban space were discussed of which only the first is discussed here: choice, convex place direction and space appropriation (Leymarie et al. 2008). As proposed by van Tonder (2004) as well as Franz and Wiener (2008), forks in structural skeletons for spatial configurations are likely to represent some kind of perceptual choice location for observers. If one assumes the structural skeleton to be an approximation of circulation then an edge can be travelled in in both directions. Because a ridge is always relating at least two polygon edges, the user arriving at a fork with 360° vision (or simply turning around) will have the choice of routes proportional to the number of edges connected to this location of intersecting axes (fork). At a fork, the routes made up of the derivative edges are equally perceptible to the user and thus he has an equal choice of moving in either direction (purely from this geometrical distance criteria, not any other way-finding parameters). Small changes in the configuration of footprint edges influence the locations at which the user either perceives no choice of path when travelling along a ridge, one choice or two choices within a given distance (Fig183).

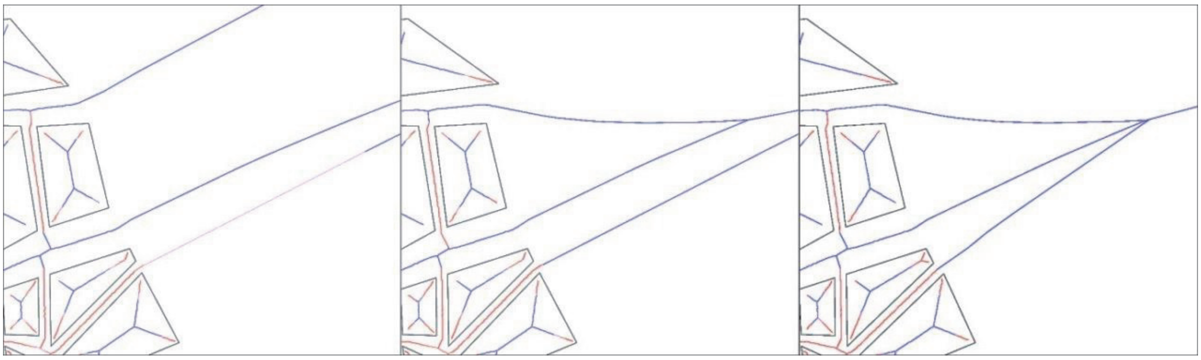


Fig183. *Straight Skeleton by CDR, 2008: showing the concept of route choice at fork locations; (left) skeletal ridges run parallel due to configuration of composing edges, which gives no perceived choice of alternative movement; (middle) one edge is slightly adjusted to produce a location at which two routes can be visually perceived in the field; and (right) another slight adjustment produces two locations where all three possible*

The Medial Axis Generator was redeveloped as a Spatial Topology Graph (STG) for the European RIBS project. The redevelopment was conducted by Anders Holden Deleuran with the author using the programming language Python to develop custom Grasshopper components in McNeel's Rhinoceros CAD environment. The purpose for redeveloping a medial axis approximation was seen by three of the strands of discussion above:

a) finding a generic partitioning model that approximates the topological configuration of a spatial layout, i.e. its permeability structure,
b) allowing for network representations beyond simple tree graphs and
c) enabling the same representation for analysis of spatial structures with graph/ network theoretical measures.

In context of RIBS this was desirable because it was assumed that building security infrastructures need to be inherent in the spatial structure, i.e. the building configuration needs to be spatially resilient to attacks (Deleuran and Derix 2013).

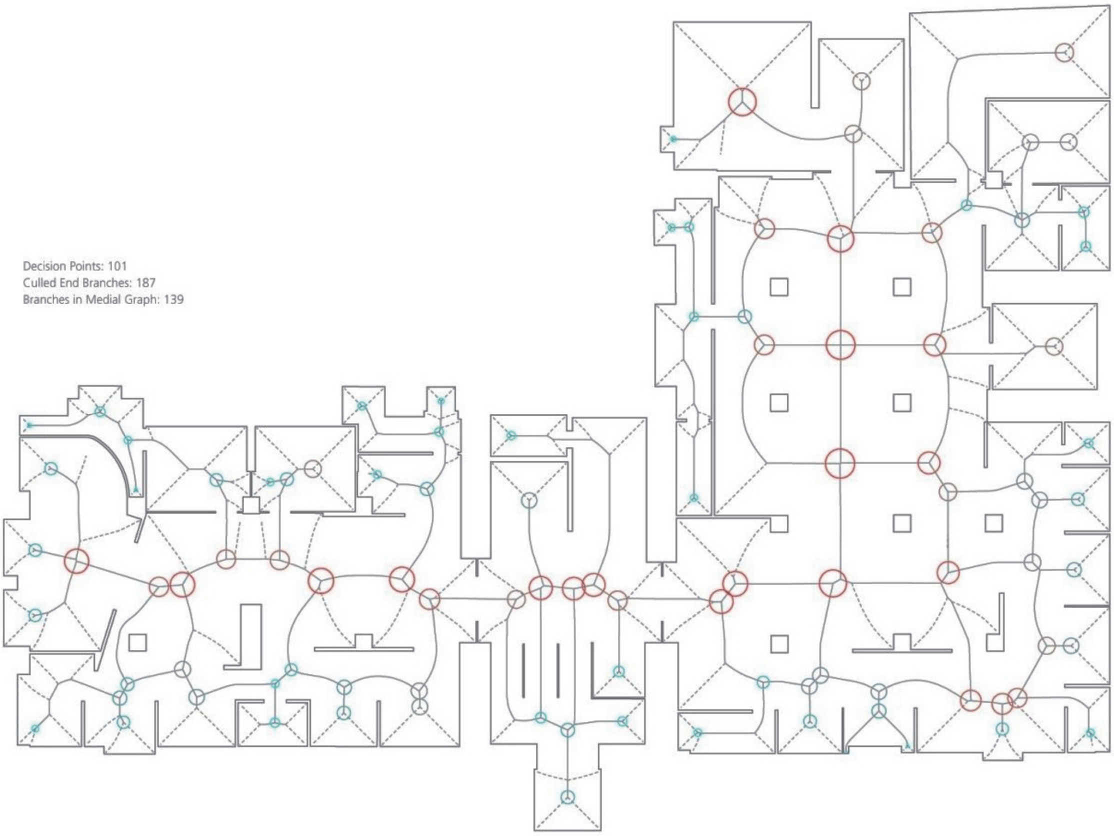


Fig184. *STG approximation of Medial Axes via Voronoi construction by CDR, 2012: the structural skeleton of a layout plan of the case study object of the RIBS project (colour of the node circles indicate the size of space area they represent)*

The construction method of the STG used here is based on the Voronoi method (see 3.4). Voronoi diagrams apply a similar principle to spatial partitioning as the intended structural skeletons by creating boundaries that are equidistant between vertices. The resulting graph can be manipulated and simplified for analysis as the Voronoi construction method produces a lot of edge vertices that might not add spatial information. But the main reduction in complexity of the resulting structural skeleton refers to a behavioural aspect: medial axes and structural skeletons produce dead-end edges with end-nodes of valence 1, i.e. single connection that emanate from concave corners. For spatial analysis those end-nodes are equivalent to Hillier's a-space types (1996; see 2.7) and only partially interesting, and for user behaviours in relation to spatial configuration they do not add information. Hence, the reduction allows to cull end-nodes and their appendix edges as well as through-nodes with valence 2 (Hillier's b-spaces) to reduce the skeleton to perceptual choice nodes only, i.e. nodes with valence >= 3.

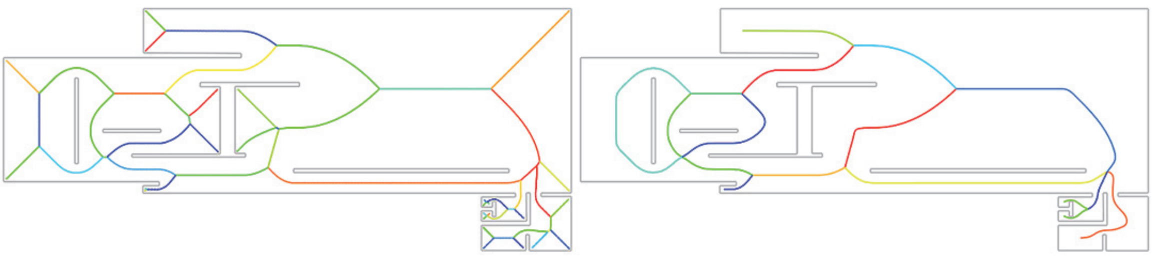


Fig185. *STG, 2012: (left) the medial axis via Voronoi construction with end-node edges into corners and (right) the pruned and simplified skeleton, which corresponds closely to Franz and Wiener's place graph (Deleuran and Derix 2013)*

Apart from providing a consistent partitioning model, the STG enables the application of graph theoretical measures to analyse spatial configurations. The previous section introduced *betweenness* centrality as graph theoretical measure for access integration of nodes within a network. For RIBS centrality measures provided information about accessibility to spatial partitions such as rooms and areas of different asset classes. Furthermore, as a measure of spatial resilience, connecting edges could be identified that would severely damage the infrastructural operations in case of attacks. The risk levels of nodes represent a control state of such nodes as already conceptually introduced for social networks and psychology by Linton Freeman (1977). The following measures were used by the STG:

- node degree (valence)
- location centrality (depth)
- betweenness centrality
- node centrality (closeness)
- graph cycles

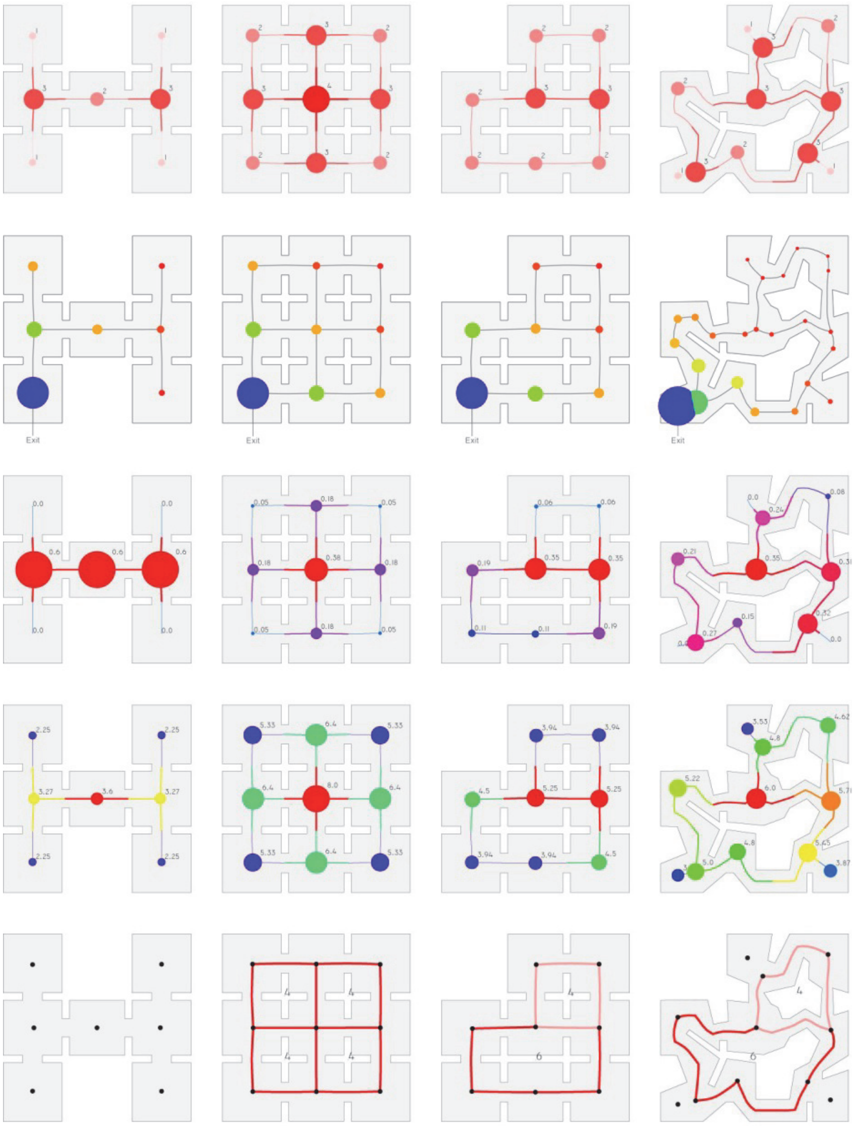


***Fig186**. STG measures on a simplified layout grid: from top to bottom: node degree or valence; location depth; betweenness centrality; closeness centrality and cycles (Deleuran and Derix 2013)*

Four of the five measures are demonstrated in Fig186 using simplified spatial configurations of various complexities as fictional layout plans. The same analysis could be done for layered layouts, such as buildings with multiple floors. The bounding polygons of stairwells are treated like flat polygons and vertices on their edges interpolated as described above. Only two landing polygons need repeating and the resulting skeletal nodes re-connected into a three-dimensional model. While this construction works for the topological graph analysis, it does not take geometrical properties like height difference into account.

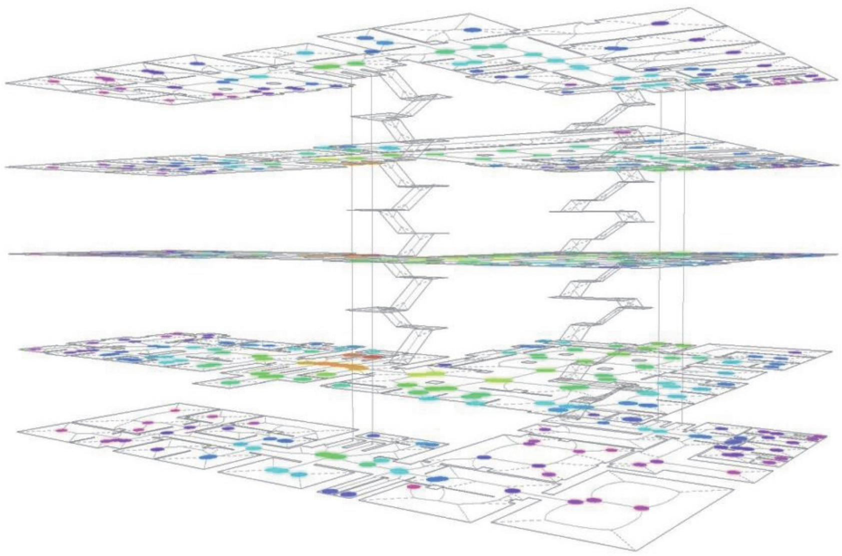


Fig187. *STG, 2012: multi-layered building layout with connecting structural skeleton running through stairwells and lifts on the RIBS case study object*

Resulting analysis values are exported as tabular data to allow further integration with other measures and models. The STG model served to integrate various spatial simulation models, such as the VPTA, which could be run on the resulting nodes of the skeletons. Eventually, STG was generalized via implementation on a series of case study building plans, which demonstrated that the method was readily applicable to any building layout in two or three dimensions.

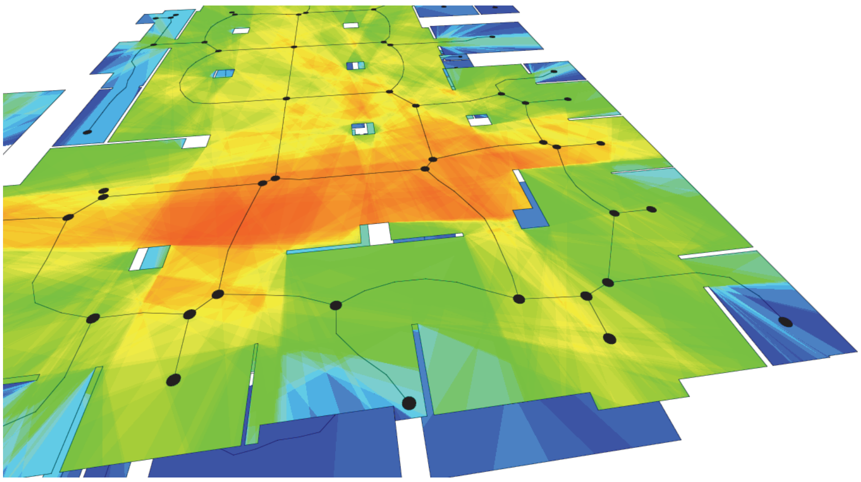


Fig188. *Visual Polygon Traversal Algorithm (VPTA) executed on the STG across multiple floors on the case study object for the RIBS project, 2013*

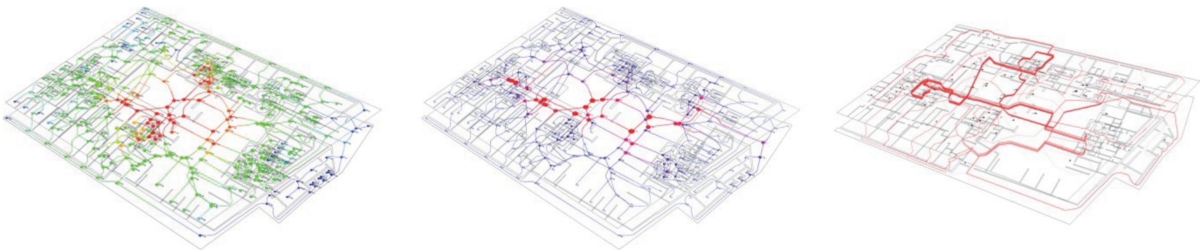


***Fig189**. STG applied for generalization to both floors of the case study National Farmers Bank in Falun, Sweden by Hultman & Holmer architects, 1973: (left) closeness centrality, (middle) betweenness centrality and (right) cycles*

The structural skeleton or medial axis representation of spatial topologies appears to solve the three issues presented above: a) consistent topological partitioning representation; b) complex building network representation and c) basis for network analysis. In fact, it enables the synthesis between Hillier's (1996) topological categorization of spatial types in p-complexes and van Tonder's (2004) medial axis model for the correlation of visual perception and location, which Arnheim (1974) based on Koffka's gestalt principles (Deleuran and Derix 2013). Nodal measures from network analysis presented here corresponds well to Hillier's topological types of space where he distinguishes between four classes of occupation or movement: a-spaces as end-spaces for functional occupation; b-spaces as link spaces between occupation spaces; c-spaces as single-ring spaces and d-spaces as multiple-ring spaces (Hillier 1996; see 2.7.2 Fig33). Those four types of movement spaces can easily be identified in the STG diagrams. It could be argued that there might be more than Hillier's categories as movement nodes occur that represent hybrids of b- and d-spaces, meaning they link sub-networks and have high control because all footfall needs to go through them; but they also lie on more than one cycle and hence allow for high flexibility and little active control.

Each type of space has an implicit risk and control or behavioural affordance value, which align well with the RIBS project's resilience criteria. For example, b-spaces represent high risk places in a network as their obstruction would break the network permeability (or building occupation infrastructure) into two sub-graphs; or simply cut off sub-spaces that are linked to that node location. People and assets trapped in the sub-graphs of spaces that have no external access are at a higher risk. Resilience therefore in a topological configuration of movement spaces means that few locations as possible should be cut off by culling a single network node. High value assets on the other hand were not meant to be placed in a highly integrated partition (c- or d-spaces) close to public interfaces but should reside deeper in the configuration (a-space) with controlled access (b-space).

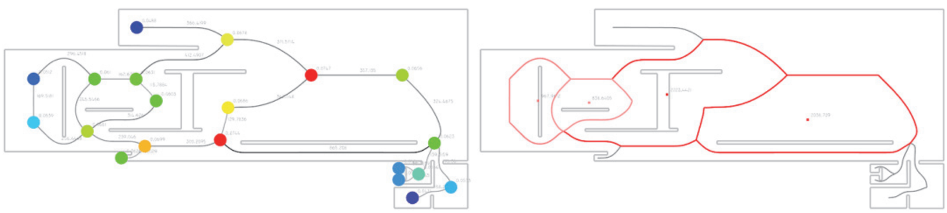


***Fig190**. STG applied to Franz and Wiener's case study(2008), showing the reduced medial axis graph (culling end-spaces or 'a-spaces'): (left) colours indicate through traffic identifying a hierarchy of places where the red nodes are positioned on two large cycles that connect two building sub-networks and therefore representing a hybrid between Hillier's b- and d-spaces; (right) the resulting cycles with thicker and deeper red representing larger cycles (Deleuran and Derix 2013)*

Conversely, in some building sectors such as workplace design, it is desirable to design in c- or d-spaces as they facilitate choice in movement and high probability of social encounter. Also in egress scenarios, planning in circulation cycles is desirable as more than one access route exists to each location. Particularly, the spatial property of perceived spaciousness represents locations with a combination of high visual choice and movement cycles, as both attributes provide the sensation of reach beyond actual access.

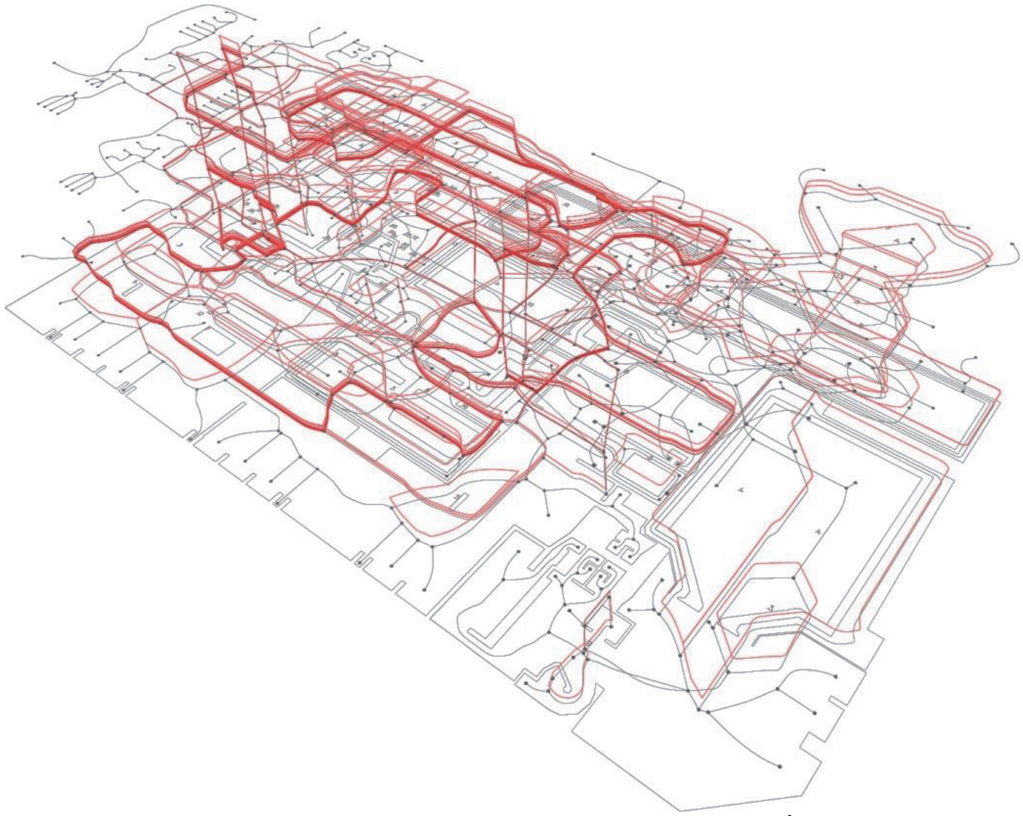


***Fig191**. STG applied for generalization to the case study Banco de Londres y América del Sur in Buenos Aires, Argentina by Clorindo Testaarchitects, 1959: the many topological cycles that exist in the movement structure with the thicker red lineweight indicating longer more integrated circulation loops*

The automatic generation of spatial topology networks should provide the ideal basis for space syntax analyses methods because a) it allows the evaluation of graph theoretical measures as done on axial maps, b) is generated from convex space partitions and c) topologically produces the hierarchical depth maps that reflect Hillier's justified graphs (Hillier and Hanson 1984). In fact, as Franz and Wiener suggest: "The minimum wall distance algorithm appears as a useful basis for well-defined place graphs encoding the spatial topology on the basis of the geometry" (Franz et al. 2008, p289).

6.2.5 Space-Behaviour Correlation

The last case study in this section discusses the use of a graph theoretical measure in conjunction with an information visualization algorithm for space partitioning of building floor layouts. In 2010 CDR was commissioned by the Fraunhofer Institute for Workplace Organization (IAO)[53] to develop an immersive and interactive

[53] www.iao.fraunhofer.de, accessed 09.11.2014

demonstrator in real-time for the Future of Construction project (FuCon), funded by the German ministry of Infrastructure, Construction and Urban Planning and managed by IAO. The purpose of the demonstrator was to provide a proof of concept platform for the FuCon concept, which foresaw the integration of all design phases via parametric modelling. Within the available budget and time, and subject to the virtual reality (VR) visualization, CDR developed three simulations of the building planning process (for full project summary see 8.3) that were mediated via the IAO's VR system called VRfx (Krause et al. 2011):

a) building massing and envelope on site
b) building programme distribution and massing
c) programme allocation on floors according to circulation

The third simulation developed by Pablo Miranda was intended to generate a building floor layout that creates a clear correlation between allocation of programme and an accessibility algorithm. The building programme was given by a hypothetical laboratory building's area schedule and the furniture grids for the lab spaces. The purpose of the demonstration was to show that such a traditionally highly-constrained layout could be generated from the permeability network (circulation diagram) in accordance with the adjacency matrix, and thus allow functional areas to be distributed as a consequence. The algorithm for generating the circulation structure borrowed from the information visualization discipline, is called hierarchical edge-bundling (Holten 2006). In information visualization edge-bundling was introduced to visually de-clutter complex connectivity diagrams by combining graph edges that share joint properties like directions into a single bundle. This method is analogous to Frei Otto's experiments with wet strings to generate lean structures sharing force edges such as minimal path networks (Otto and Rasch 1995) which he later deployed to investigate movement networks (Otto 2008).

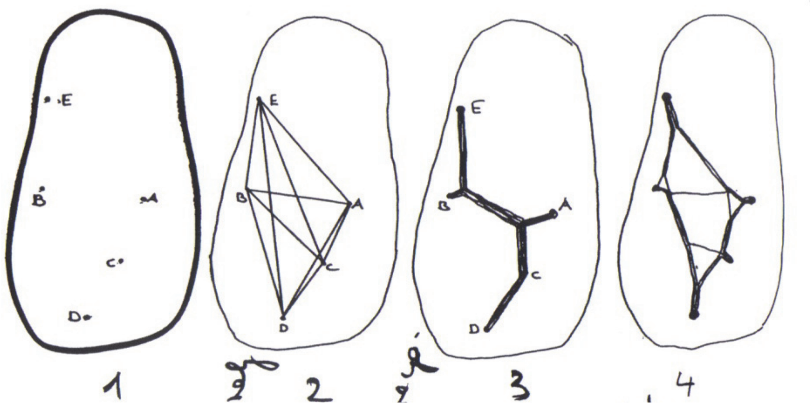


Fig192*. Frei Otto, 2008: path network definitions from nodes (1) and all adjacency connections (2) from which a minimum spanning tree (3) and a bundling network (4) is generated*

By aligning connections such as movement links, areas of partitions are increased that can be occupied by the programme. Alignment in hierarchical edge-bundling takes preferred adjacencies into account and depending on the allocation of initial programme nodes, resulting bundled movement links are concentrated between nodes that are by preference more integrated due to larger numbers of connections. The weakness of this approach currently is the initial allocation of programme nodes, which are manually placed (or interactively moved).

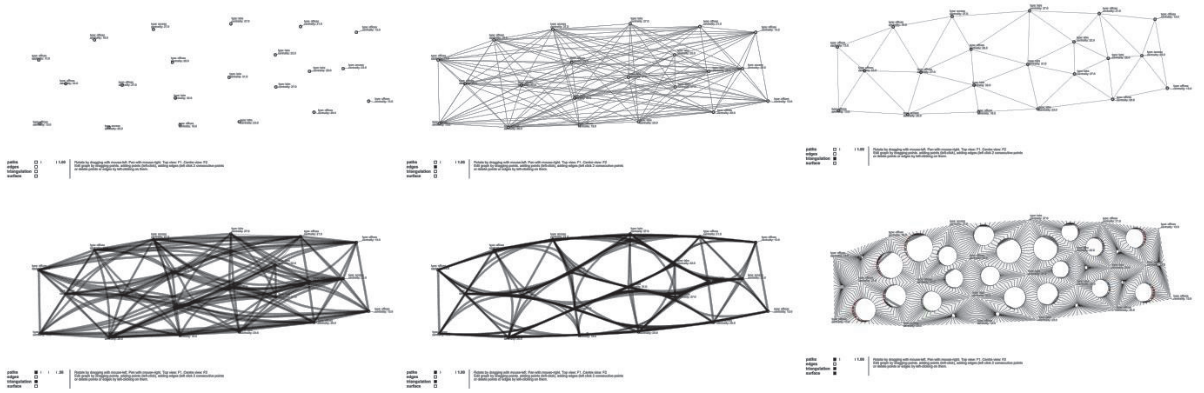


Fig193. *FuCon edge-bundling for building programme layout, 2010: the construction sequence showing from top left: the programme nodes; the adjacency connections; the Delaunay triangulation mesh; (bottom) the adjacency edges bundling towards the mesh edges at 50% strength; at 100% strength with emerging programme areas and finally the building envelop with atria*

The eventual model represents a hybrid between algorithms of Holten's (2006) hierarchical edge-bundling, Qu's and colleagues' control-mesh edge-bundling (Qu et al, 2006) and a bespoke heuristic for force-directed edge attraction. First, all programme nodes are connected with a link according to the area schedule's adjacency matrix and a constrained Delaunay triangulation produces a mesh between the nodes (Delaunay 1934). The mesh provides a topological control structure for the adjacency edges to be attracted to. The adjacency edges are sub-divided by a fixed number and new vertices are spread evenly across the length of the edge. The interpolated vertices are moved toward the nearest Delaunay mesh edge, resulting in the bundling of the adjacency edges along the Delaunay mesh edges according to a force that can be set in the GUI.

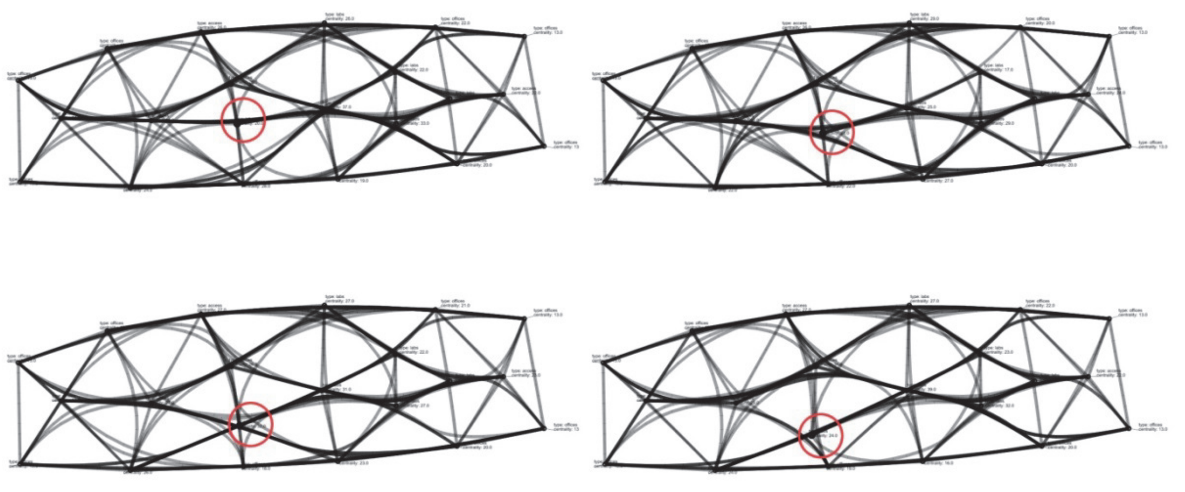


Fig194. *FuCon edge-bundling for building programme layout, 2010: four minimal interactive adjustments of one central node resulting in larger changes of circulation and area patterns around that node with strong differences in the centrality values (top left counter-clockwise: 20 / 24 / 24 / 32)*

Betweenness centrality is applied to each programme access node that reflects a hybrid between the adjacency specification and the Delaunay mesh edges. The mesh edges provide the integration from movement links but the adjacency specification provides information about hierarchy. The betweenness centrality is thus a synthetic measure between the two: movement and hierarchy. The centrality measures the performance of the algorithmic interpretation of the adjacency matrix and the partitioning, while the edge-bundling visualizes the integration of a programme node based on both edge integration properties.

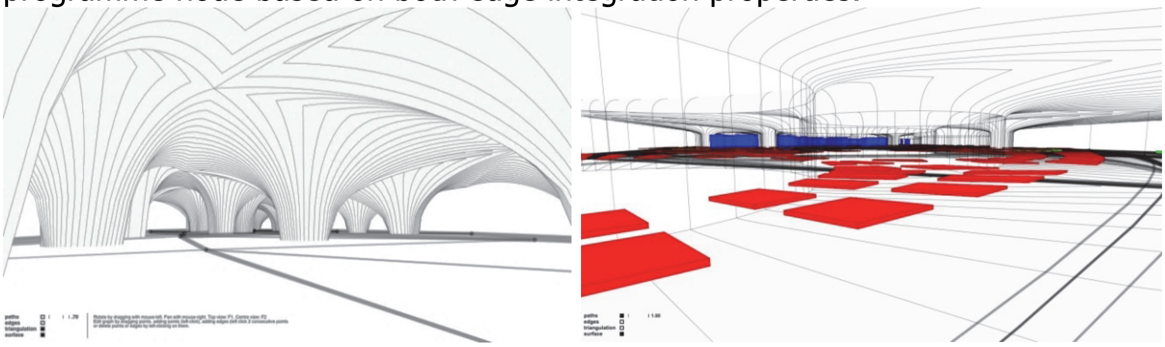


***Fig195**. FuCon edge-bundling for building programme layout, 2010: (left) the surface envelope and atria being placed across the emerging areas (see atria at dependent on the triangulation); and (right) the furniture grid being inserted into the allocated programme areas*

Emerging area partitions are bounded by three programme nodes each, which fill the partitions with their diagrammatic furniture grids. That way the circulation movement edges linking programme nodes are always adjacent to the programme areas they facilitate access to and align with the assumed spatial hierarchy. Atria are inserted at the centre of each partition to allow natural daylight to penetrate into the workspaces: the larger a partition, the larger the atria. The architectural concept for this simulation was based on Sanaa architects' Rolex Learning Centre at EPFL in Lausanne[54], which was designed to function as a single floor library where programme and atria are a function of the circulation infrastructure.

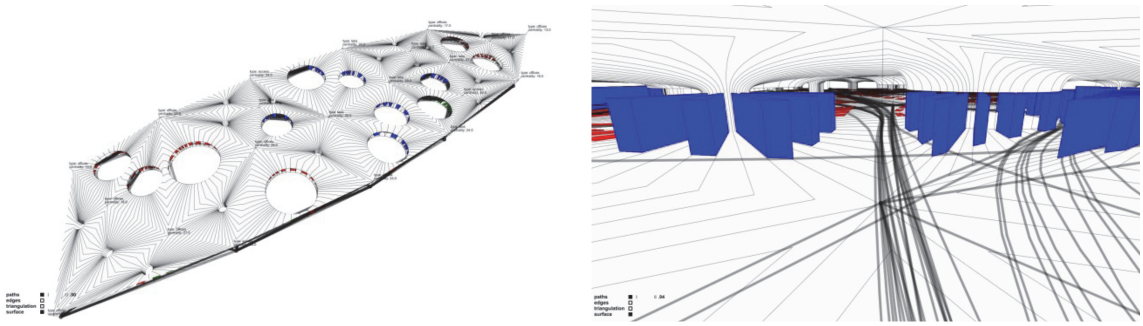


***Fig196**. FuCon edge-bundling for building programme layout, 2010: (left) atria surface; (right) ceiling sloping towards ridges above circulation*

The FuCon programme allocation by edge-bundling model creates a strict correlation between movement and programme. The movement concept applied here is not simply based on shortest geometric routes but fuses perceptual properties of networks and visual legibility with access behaviour and area allocation constraints. The eventual placement of court yards around the centres of the emerging areas are themselves scaled according to proximity to the circulation network, which in turn is

[54] http://rolexlearningcenter.epfl.ch , accessed 09.11.2014

concentrated on major flows. The diagrammatic walls-turned ceilings slope upwards towards the circulation ridges of the flow centres. This proximity with flow integration is reminiscent of the study done by Franz and Wiener on Proxemics within space and the perception of wall and column distances: the most spacious areas – perceived and actual - run along equidistant ridges where circulation with highest footfall is proportional to the corridor widths.

From a professional practice point of view, this methodology of allocating building programme is the opposite of the standard area efficiency approach, where areas are packed on geometric evaluation criteria and behavioural performances such as movement and perception are used retrospectively to refine the geometric layout afterwards.

6.2.6 Summary of Performances and Measures of Behavioural Graphs

This section introduced graphs as behavioural diagrams and partitions as the spatial aggregations that graphs are calculated from. While maps in the previous section produce perceptual values at discrete positions that are analogous to the mapped territory, graphs abstract the territory into topological formalisms that represent behavioural affordances. Those affordances are calculated from geometric ratios of spatial configurations and inform both global structural as well as local spatial performances. All graphs are based on some calculation assumptions about the aggregate spatial elements they represent, called partitions. The partitions and graphs discussed in this section have emerged as principal spatial representations for human-centric environments, representing basic associations of user-occupant behaviours for spatial design as well as design heuristics relating to spatial planning. The table below correlates the formalisms – graphs, partitions and network measures - that abstract spaces to the encoded behavioural associations and lists planning aspects of spatial environments that they can be applied to.

The table lists the edge-bundling in the Connections category, i.e. graph formalisms section although it is strictly speaking not a graph representation but a transformation of topological graphs. Additionally, the structural skeleton or medial axis is listed under Partitions and correlates to many circulation design aspects of multiple design objectives, although structural skeletons do not strictly speaking represent movement structures. They only approximate movement structures in that they generate centre-line configurations between edges and locations that often coincide with movement ridges.

Generally, it can be observed that maps are a first order representations of space by processing a context *inwards* onto positions. Because they are local, they cannot be transformed in isolation from their environment. Graphs on the other hand, are second order representations of space by generating dependencies *outwards* from local aggregations to global configurations. Thus they encode syntaxes that allow the generative transformation of global configurations, which give value to local positions. Local positions are resultants from the ratios between geometric elements that create aggregates for graphs.

		STRATEGIC PLANNING								MOVEMENT STRUCTURE							PLACE MAKING					SIZING		
DESIGN OBJECTIVES / PERFORMANCES *correlate to* SPATIAL FORMALISM	BEHAVIOURAL AFFORDANCE	workplace organization/ groups	face-2-face informal communication	logistics	operational organization	capacity evaluation	asset management	security measures	risk management	circulation design	way-finding	flow distribution	place activation	neighbourhood hierarchies	accessibility (walk/ cycle)	access control (social/ program/ security)	appropriation of space / private use	public space	program allocation	land-use allocation	facility allocation	street/ corridor widths	street-elevation aspect ratio	density and scale
CONNECTIONS																								
Shortest route (directed graph)	quick movement, direct communication	X		X			X	X			X	X	X		X					X				X
Direct route (least angle) (directed)	simplest movement / layout zones	X	X	X	X			X			X	X	X		X					X				X
Bi-directional routes	simultaneous flows	X		X	X		X	X	X	X		X			X	X				X				
Spannning Trees	connecting places / general movement			X	X		X			X		X	X										X	
Bundling	movement hierarchy/ legibility		X		X					X	X	X	X		X	X		X	X		X	X		X
PARTITIONS																								
Area structures/ Tree maps	organization hierarchy	X		X	X	X	X																	
Convex partitioning	enclosure, belonging to	X						X			X					X	X			X				
Distance field	proxemics / belonging									X	X		X				X	X	X				X	X
Structural skeleta / medial axes	orientation / movement		X	X				X	X	X	X					X	X	X				X	X	
Branching / Forks	choice location		X					X		X	X	X	X		X	X		X			X			
Edge alignment / proximity	static activity / directionality	X										X	X					X	X	X	X			
NETWORK MEASURES																								
visual depth (spot)	group belonging/ exposure	X	X		X		X	X	X	X	X	X	X		X	X	X	X	X	X				
spatial depth (location centrality)	remoteness/ privacy	X		X	X	X	X	X		X	X	X			X				X					
node connectivity (valence)	compactness/ dependence	X		X	X	X	X	X				X				X		X	X					
betweenness centrality	influence/ control		X		X		X	X	X	X	X	X	X	X	X	X		X	X	X	X	X	X	X
closeness centrality	proximity / integration	X		X	X		X	X	X	X		X	X	X	X	X	X		X	X				
cycles	spaciousness/ choice		X	X	X	X	X	X	X	X	X	X	X	X	X	X		X	X	X				

Fig197. *Formalism-to-Behaviour-to-Objectives table of relations*

In a commercial setting, the use of graphs and graph theoretical performance measures to analyse and generate KPIs is new for spatial planning, although graphs were introduced as some of the earliest mathematical abstractions for spatial analyses (Alexander 1964). Disjointed efforts have been made by Space Syntax ltd and others to introduce graphs into the design process but as those developments are isolated from design and limited to strategic consultancy, they have not been standardized yet into planning guidance or compliance requirements. As an increasing amount of graph-based design support software is coming to market[55], it is only a question of time before the use and interpretation of graph-based analysis will become an accepted standard.

6.3 CONNECTIONIST SPACE | ASSOCIATIVE FIELDS

The previous two sections discussed firstly maps as observer-external representations of perceptive properties of place and secondly graphs as observer mediating behavioural diagrams of configuration. Finally, a third order representation is introduced that aims to understand and instrumentalize the observer's cognitive assumptions about properties of configuration. Reversing the agency of the observer, the observer becomes the agency of the model. Maps represent the observer in space, graphs represent the space of the observer and finally associative networks represent the space inside the observer (internal states).

To decode the cognitive organization of a spatial environment the previously discussed representations serve as inputs to generate comparative classifications of

[55] Such as ESRI's GIS platform for urban planning: http://www.esri.com, accessed 15.08.2014

spatial properties. While maps calculated on a discrete field analogous to the territory and graphs evaluated a formalized geometry of the territory, associative networks compare generated intensity values and configurational syntaxes as normalized numerical input data quantities, removing the territory from representation altogether. Creating associations means to find the differences and similarities between mapped locations and generated configurational formalisms. From this comparison of places and configurations, profiles are established that describe the set of properties by which an observer associates spatial types. In other words, a quasi-experiential or intuitive description is generated from associative networks. Richard Coyne called those profiles *schemata* that underlie *episodic* design, which means to design from narratives founded on associations with places: *"A particular experience, such as entering a restaurant, may trigger the recollection of a general restaurant experience. […]The restaurant schema may contain a description of the expectations attached to the setting. There are patterns about what to expect and how to respond."* (Coyne and Newton 1990, p39)

Following the discussion in 2.4 Associative Reasoning, associative networks work without a definition of syntactical rules and therefore omit the specification of ontological organization for spatial environments. Spatial profiles by association emerge during the comparison process, when associations between places and configurations are learned: "*There is no explicit representation of a schema. However, a schema is implicit in the pattern of associations generated by the system during the learning process"* (Coyne and Newton 1990, p40). Associations are free to form from the data sets provided and the benefit of artificial learning is that relationships can be found between apparently disparate properties, which can describe bespoke experiences.

		dove	hen	duck	goose	owl	hawk	eagle	fox	dog	wolf	cat	tiger	lion	horse	zebra	cow
is	small	1	1	1	1	1	1	0	0	0	0	1	0	0	0	0	0
	medium	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0
	big	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
has	2 legs	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
	4 legs	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
	hair	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
	hooves	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
	mane	0	0	0	0	0	0	0	0	0	1	0	0	1	1	1	0
	feathers	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
likes to	hunt	0	0	0	0	1	1	1	1	0	1	1	1	1	0	0	0
	run	0	0	0	0	0	0	0	0	1	1	0	1	1	1	1	0
	fly	1	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0
	swim	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0

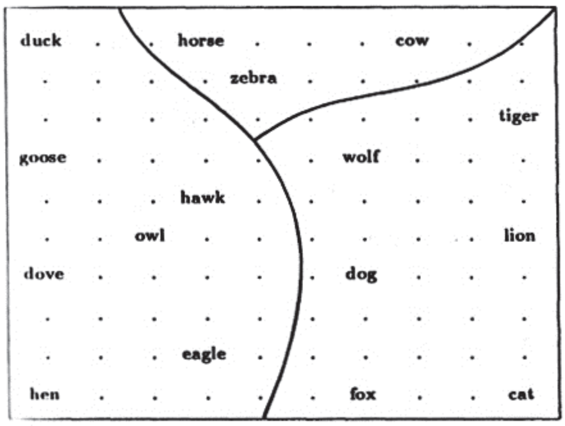


Fig198. *Self-Organizing feature Map (SOM) by Teuvo Kohonen (1995): (left) the input data defining animals by features using binary encoding (true/false); (right) the output classification map (SOM) relating animals spatially into 'perceptive fields'*

Models discussed in this section are based on artificial neural networks (ANNs). ANNs are models that form part of the conceptual class of *connectionism*, which as the name suggests are models that process information through connections between data units. Connectionism was originally developed to investigate cognitive processes using artificial intelligence as a vehicle (see 2.4 Associative Reasoning). As discussed in 3.5, there are two main categories of ANNs: supervised and unsupervised networks. Unsupervised networks self-organize input samples to output classes that are not pre-determined and thus generate associations between input features and output classes. Those output classes when organized spatially are also called *perceptive fields* by Teuvo Kohonen (1995), who invented the self-organizing feature map (SOM).

6.3.1 Experiencing Movement : Space-Action Co-Respondence

Research into associative networks for spatial design at CECA began in 1999 during the author's MSc Computing & Design (Derix 2001). The concept of connectionism was raised from the results of a movement study that mapped repeated walks along an identical metric route on plan (the Canary Wharf shopping mall from Cabot Square to One Canada Square). In plan the assumed walking path would be a straight line across a distance of 150 meters, which would take approximately 110 seconds or two minutes at an average walking speed of 1.4 m/s. Each walk was recorded verbally monitoring all actions along the path, which would either deviate from the average walking speed or centre-line direction. The resulting series of recordings was mapped into plan diagrams by a notation convention, attributing graphic elements to movement actions (Derix and Jagannath 2014a)

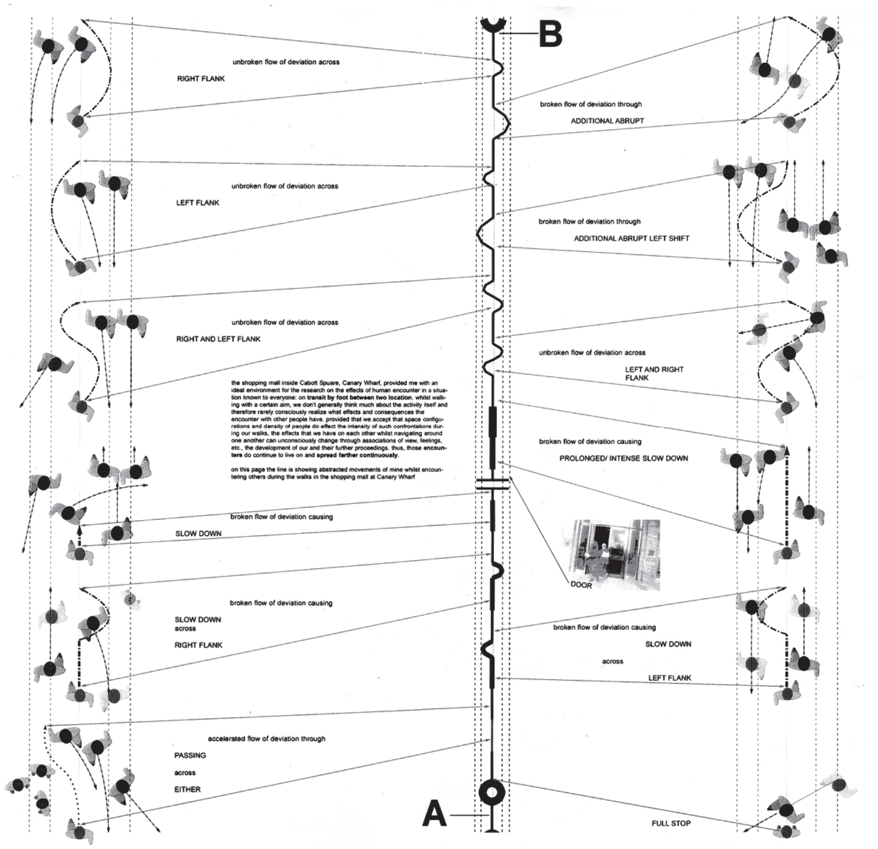


Fig199. *Walking maps, 1999: the graphic convention to translate manual and audio recordings of walks into normalized maps; at centre, a demo constructed diagram interpreting different hypothetical events during a walk*

The resulting movement maps showed that all walks deviated from the assumed metric standard, because of unforeseen events mainly responding to other people who in turn respond to yet other people. All people are simultaneously co-responding within a spatial and a time-based operational frame. As proposed by the sociologist Bruno Latour's (1987) actor-network theory (ANT), none of the elements of a heterogeneous system take command in the generation of an empirical situation and are thus operationally equal, meaning that people do not dominate space or time.[56] This situation pointed towards an autonomy of a system where people and space are interacting outside the conventions of metric representation.

Additionally, the mappings revealed co-occurrences at locations where certain (generic movement) actions were more likely to occur at certain times, which generated classes of associations between occupation, place and time. Using Coyne's words, an experiential mall schema was approximated for this particular location as a potential phenotype for the genotype mall.
The discrepancy between standard metric representation and actual events within an apparently autonomous system aligned well with connectionist models that are based on properties of complex systems such as non-linearity, openness or distributedness. Paul Cilliers (1998) highlighted the isomorphism of ANNs and complex systems such as language or architectural space in his book Complexity & Post-Modernism and led the author to investigate self-organizing neural networks.

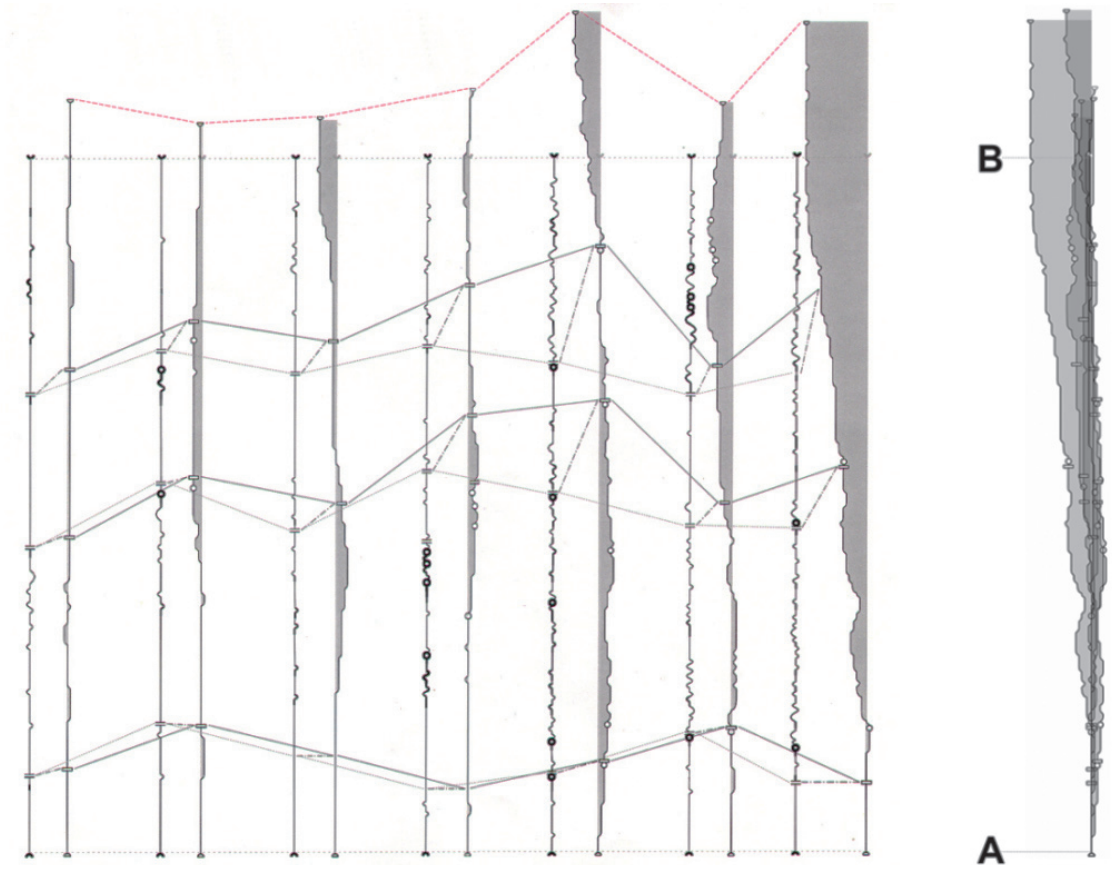


Fig200. *Walking maps, 1999: (left) seven walks 'unfolded' with multipliers for actions illustrating actual final destination locations and therefore time; (right) overlay showing how each walk varied in final destination*

6.3.2 Autonomous Spatial Cognition

The walking maps led to the research brief of developing a spatial system that could autonomously organize spatial information and generate associative schemata of space independent from human cognition. The author followed Cilliers'

[56] Latour did not want his ANT to be visualized spatially, i.e. through the lens of one of the acting elements, prefering topological representation (Latour 1999)

recommendation of connectionism as an appropriate concept to explore complex spatial systems. As discussed in 3.2, supervised networks adapt their internal connections to learn a causal correspondence between an input sample and a desired target pattern while unsupervised networks have no targets for learning but generate classes of similarity from input samples. Classes are based on differences between features, which are encoded numerically by input vectors. Because the make-up of features in input vectors can be inspected, the unsupervised classification system reveals what makes schemata similar or different not by labels or rules but simply by feature composition and weighting. Such a system was assumed ideal to cognitively organize spaces non-metrically and generate a parallel epistemology of space.

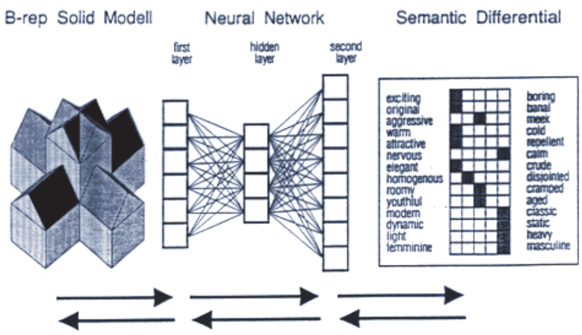


Fig201*. Petrovic and Svetel (1993): a mapping of form to semantic labels using a supervised Parallel Distributed Processor (PDP)*

The concept itself was anticipated by John Frazer in his book Evolutionary Architecture where he lists neural networks as a model in his Generative Toolbox to recognize spatial patterns autonomously (Frazer 1995, p26). Ivan Petrovic and Igor Svetel (1993) proposed a distributed automatic design system and like Coyne used Rumelhart and McLelland's PDP (1986) to generate semantic shape generators. But all precedents – Coyne, Petrovic and Svetel as well as Frazer – used supervised networks with set target patterns to learn.

SELF-ORGANIZING MAP TO SELF-ORGANIZING SPACE

The unsupervised model chosen for the autonomous spatial cognition project by the author in 1999 was based on Kohonen's Self-Organizing feature Maps (1995). The SOM represented one of the two most popular self-organizing neural networks in 1999, the Hopfield network (Froehlich 1996) being the other most common model. The SOM was preferred for its flexible input format, being able to process real numbers rather than Hopfield's binary input format. From an architectural perspective, Kohonen's SOM also presented a visually more accessible representation of the classification by literally mapping the learned classes into a two dimensional lattice. Kohonen even anticipated using the Voronoi diagram for data visualization by applying it to his maps for finding boundaries between classes.

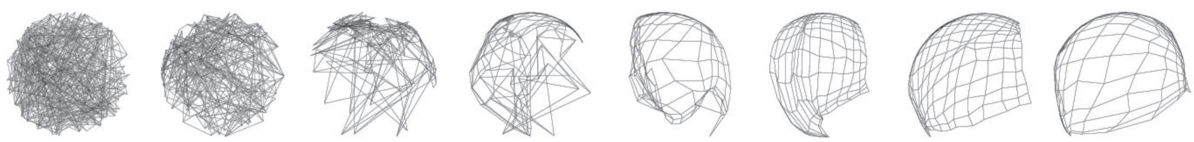


Fig202*. SOM, 1999: eight learning steps from an early experiments by author using Kohonen's SOM to map input vertices from a random distribution on a sphere; the SOM learns to correctly map the topology of the vertices across the sphere generating a 3D surface*

Initial validation of the basic SOM algorithm mapped three dimensional input into two dimensional surfaces, which are akin to minimal surfaces such as Otto's soap film models (Otto and Rasch 1995), because the surface stretches all 3D input points through a 2D lattice. The input vectors consisted of three-dimensional vertex information representing a vector space. Input samples described purely the three coordinate quantities and thus represented only one spatial dimension. All other higher-order representation of space such as edge or surface (two dimensions) and volume (three dimensions) were avoided to allow for autonomous spatial interpretations. To visualize spatial clusters of vertex densities, the surface SOM was extended to a spatial SOM, which eventually gave the project the name Self-Organizing Space (SOS). The SOM like all neural networks conducts dimensionality reduction, with an n-dimensional input vector space mapped into a representation of dimension < n. Therefore, the SOS was initially used as a density classifier and for cluster visualization rather than dimensionality reduction, since the spatial clusters had the same dimensionality as their input vectors. This initial equal-dimensionality mapping helped to develop the three-dimensional structure and its geometric embodiment.

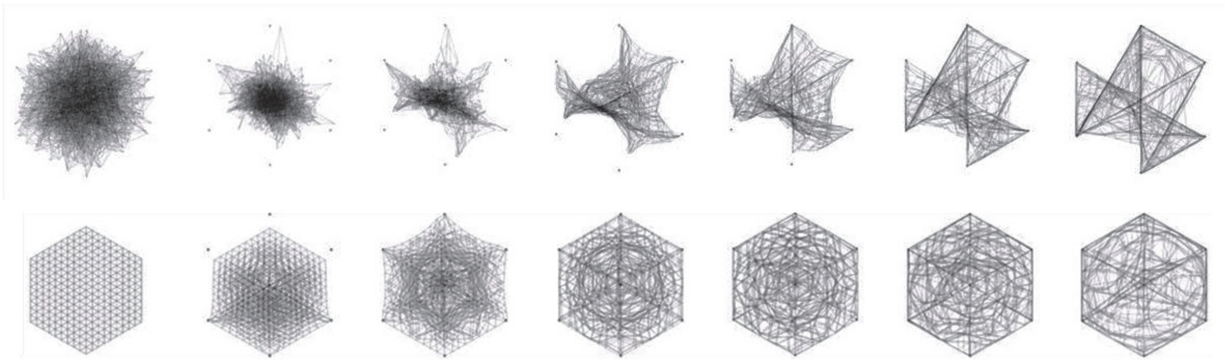


***Fig203**. SOS, 1999: the 3D lattice used by the author to map 3D vertices across a cubic input space; only the eight corner vertices were used and the initial set-up conditions of the SOS nodes either randomized (top) or orthogonally spaces (bottom); unlike smaller sized networks, two different interpretation of the same cubic space are generated*

To test the three-dimensional cognition performance of the SOS, an experiment was conducted: how many nodes does the network require to be able to 'perceive' more than one output state from different initial conditions (Derix and Thum 2000). The key insight showed that a low number of network nodes would not be able to classify differences since feedback between nodes would always be distributed to a limited number of nodes and no differentiation was possible across the map. The minimum size of a SOM depends on its application and the size of the input space. The exact size can only be found through trial-and-error, adjusting the upper bounds until the map performs without too much redundancy (which is identified by unattributed nodes 'lost' in output space). The experiment showed that an adequately sized and calibrated SOS would be able to learn different representations of a simple cubic eight vertex input space with the initial nodal set-up either randomized or orthogonally laid-out similar to the input space. While the reduced size map would always generate the same output organization under varying initial set-up conditions, the more complex network would show differences in learning. A diverging representation of the input space from the expected observer schema was important as it would provide the evidence for alternative yet rational perceptive states.

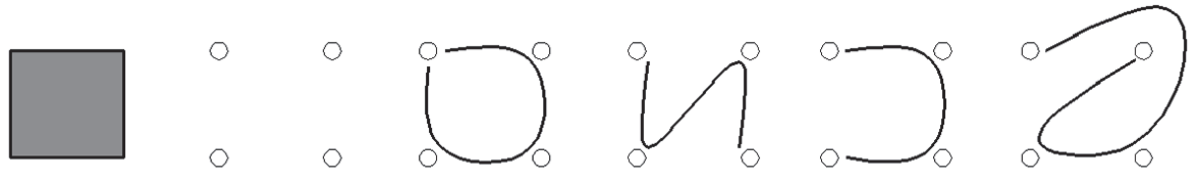


Fig204. *Schema: a drawing by the author showing how the organization of four vertices laid-out orthogonally in two dimensions would generally be described as a square through a line; but many other interpretations are possible*

To visualize the clustered nodes of the SOS morphologically, an implicit surface model was applied to the clusters, called the *marching cubes* algorithm (Lorensen and Cline 1987). The marching cubes algorithm subdivides the output space into voxels and determines if a node of the SOS lies inside or outside a threshold set by the user. The threshold represents an *isosurface* at which all nodes have equal distance to the outer surface, wrapping the clusters into spatially distinct enclosed volumes that would be equivalent to the *probability density distribution* of the input vector space (Derix 2004). The fidelity of the geometric embodiment to the spatial clusters hence depends on the size of the voxels and threshold level. The larger the threshold the more contiguous a volume becomes and vice versa, the smaller the more fragmented.

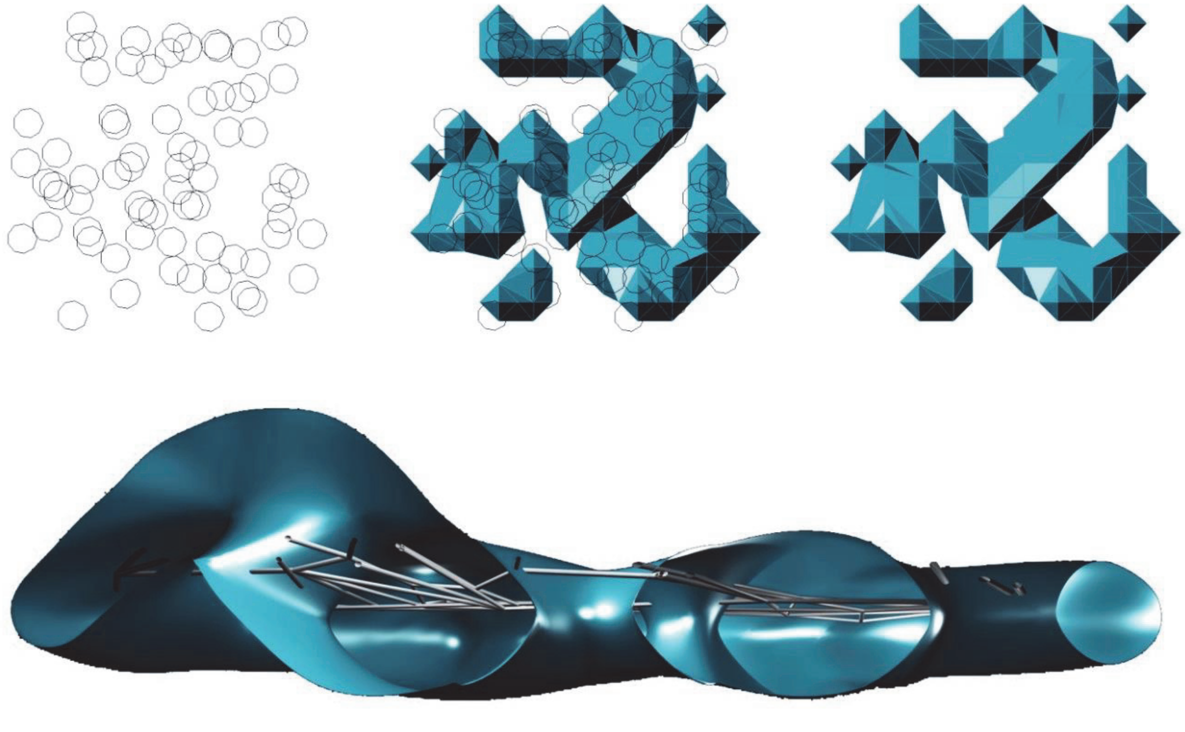


Fig205. *Geometric Embodiment, 2001: to visualize the vertex clusters generated by the SOS, an Isosurface or Marching Cubes Algorithm (Lorensen and Cline 1987) was used which produces a wrapping skin based on densities of points that reflect the 'probability density function'*

EXPERIENCING AN URBAN SITE

The SOS was eventually applied to an urban site to gather spatial data and classify each location according to the collected data (Coates et al. 2001). The purpose of this application refers back to the initial aim that a spatial cognitive model should collect and organize data autonomously. Therefore, the input samples should not be pre-selected by some guiding schema but the model should select its own input samples, as if experiencing a site.

Some additions were coded into the SOS that adapted the SOM further for spatial application: a) perceptive reach bias at nodes and dead-end halting function, b) independent generation of interpolation data and c) forgetting. In order for the SOS to collect its own spatial data, the entire vertex vector space of the urban site model is provided as input space. From this complete set of vertices, a sub-space is collected by the model's nodes by searching for vertices within a radius proportional to the geometrical size of each node's topological neighbourhood. If the neighbourhood proved inactive over several generations (no node adaptation due to lack of new input vertices), the perceptive reach radius would grow incrementally to a maximum length of half the network diameter (Fig206). If over a set number of generations[57] no new input could be found at maximum perceptive reach, the SOS invokes the halting function and terminates learning. Spaces where the model terminated were considered dead-ends and the network would 'die of boredom' to paraphrase John Frazer.

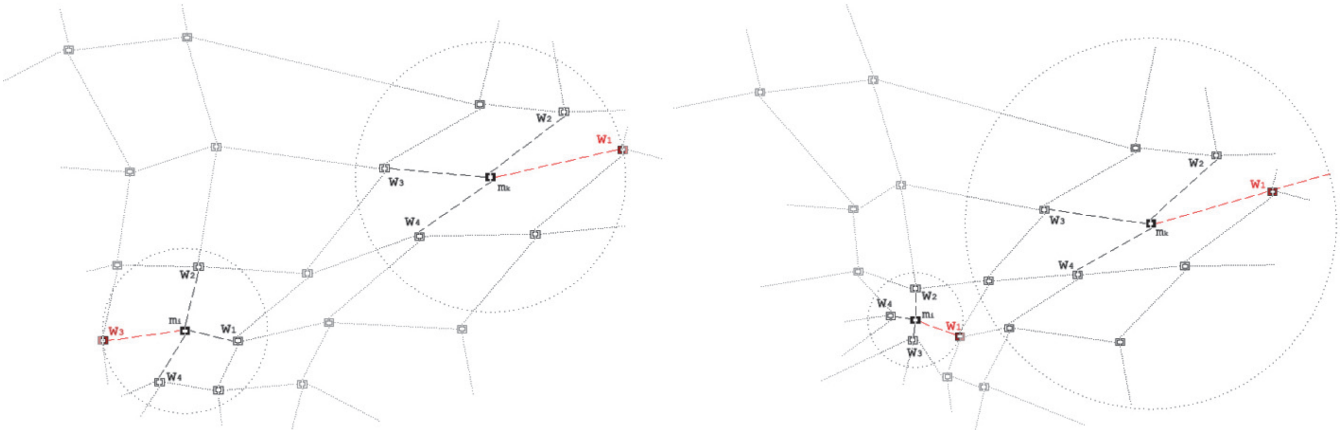


Fig206. *Perceptive Reach Bias on the SOS nodes, 2001: each node (here nodes w1 and w2) in the 3D SOS selects its own input samples for which it adjusts the radius of search after each generation of learning; the image shows (left) a node's bias being adjusted via the longest connection to its topological neighbours or (right) if no change occurs in the topological neighbourhood, the bias will grow*

Additional perceptive autonomy was built into the model by providing the nodes with an edge vertex interpolation function (Derix 2004). For each 'perceived' input vertex an edge adherence test was conducted that evaluated for closer possible vertices along an edge by projection. This proximity projection and interpolation method enabled the model to implicitly differentiate between edge and corner.

[57] A generation corresponds to a training or learning cycle

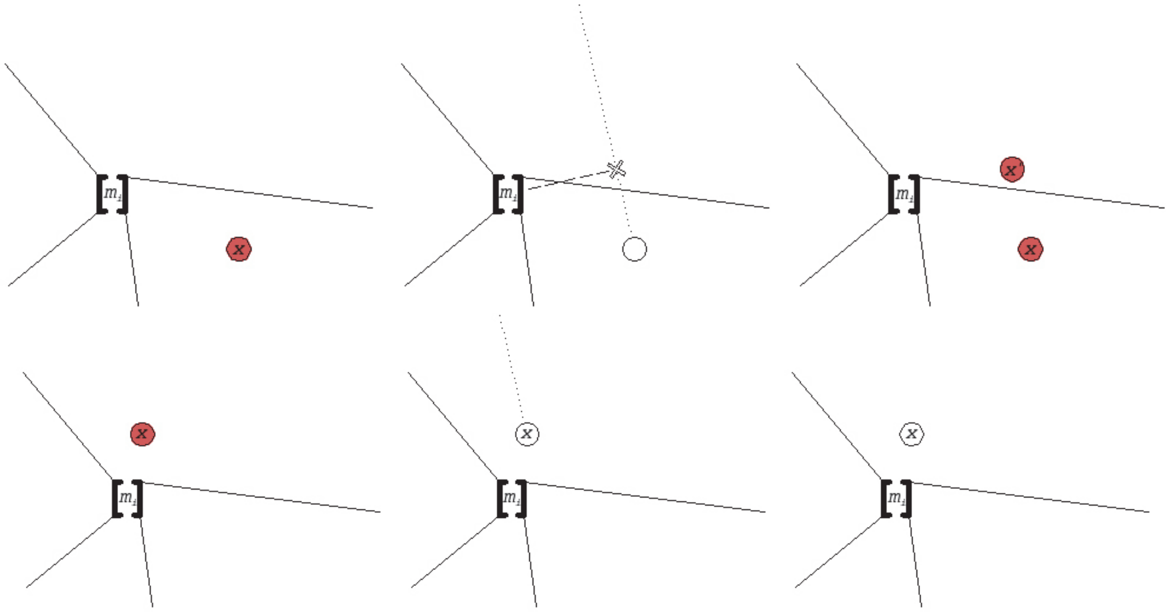


***Fig207**. Vertex Interpolation on the SOS nodes, 2001: when perceiving an input sample (vertex = x), a node ([m]) can check whether the sample belongs to a larger geometry by finding an edge it belongs to; if a projection onto this edge results in a geometrically nearer vertex, an additional sample is interpolated into the input set; this would be perceived as a 'next to' vertex; otherwise, if no nearer projection is found, it is a 'corner' input vertex*

Finally, the 'forgetting function' was necessary to maintain adaptability of the network's synaptic weights over generations. At the end of each generation, the SOS resets its learning parameters and empties the input samples list. However, because the weighting on the synaptic connections would be passed onto the next learning cycle, the SOS would take the previously learned space as a condition to learn new patterns. Compiled weightings over generations embody the 'experience' of the model and constrain its perception. This is what the theory of autopoiesis calls *structural determinism* and underlies complex systems such as language (Maturana 1978). In cybernetics Heinz von Foerster called this circular process *second order cybernetics* and proposed *that "a change in the chemical concentration of an agent in the immediate vicinity of the sensing tip, and 'perceptible' by it, causes an instantaneous contraction of this unit. The resulting displacement of this or any other unit by change of shape of the animal or its location may, in turn, produce perceptible changes in the agent's concentration in the vicinity of these units which, in turn, will cause their instantaneous contraction, etc. Thus, we have the recursion: change of sensation = change of shape* (von Foerster 1984, p295).

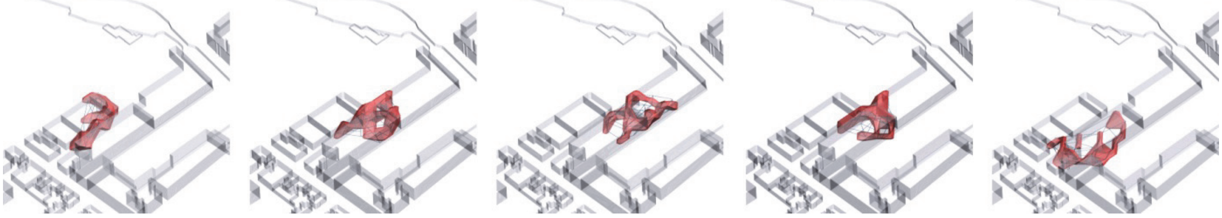


***Fig208**. SOS, 2001: five consecutive classification generations showing the morphological interpretation of the organized network moving (jumping) along and between two building volumes*

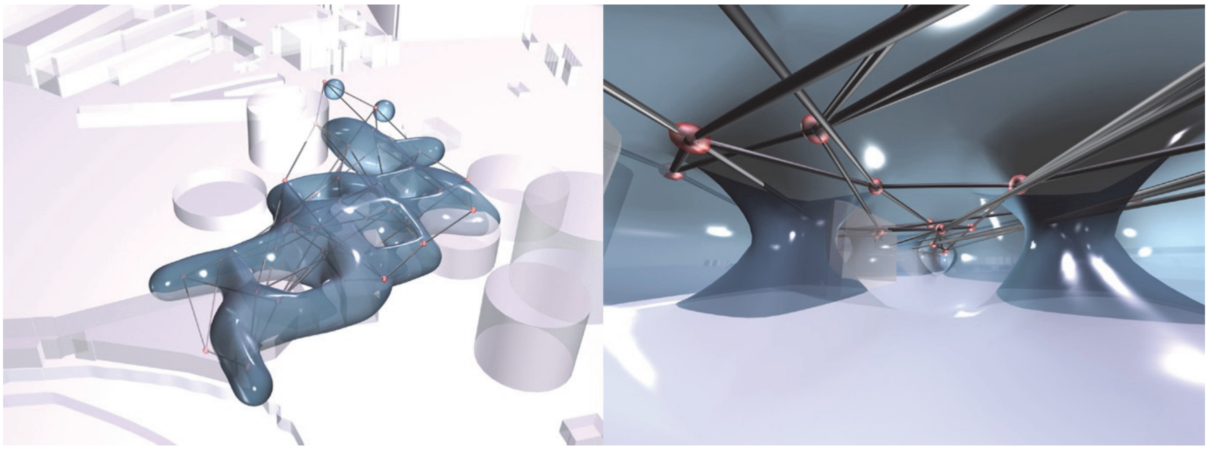


***Fig209**. SOS, 2001: (left) an instance of a rendered isosurface of a learned input space at the end of one generation; (right) an interior view of that volume showing an interesting quasi-topological morphology*

The self-selection method developed for SOS produced a quasi-living model in that the ANN literally roamed across the urban site model autonomously and from an observer's perspective *with purpose*. The fluidity of perceived movement was dependent on the visualization update rate of both network and surface in relation to the learning process. When visualizing all classified spaces across a run until the halting function was invoked, patterns of 'interesting' locations on site were revealed that were equivalent to the activation history of the network. Locations on site that appeared 'interesting' to the network represented places with higher vertex densities, triggering more intense network activation. At those information intense locations also the perceptive reach function of nodes invoking proximity interpolation was activated more often due to higher complexity of the site geometry. Emergence therefore occurs less as first order emergence of unpredictability than as structural determinism of both perceptive history and perceived structure. An isomorphic mapping is established between the perceptual conditions of the site and the behavioural activity of the network.

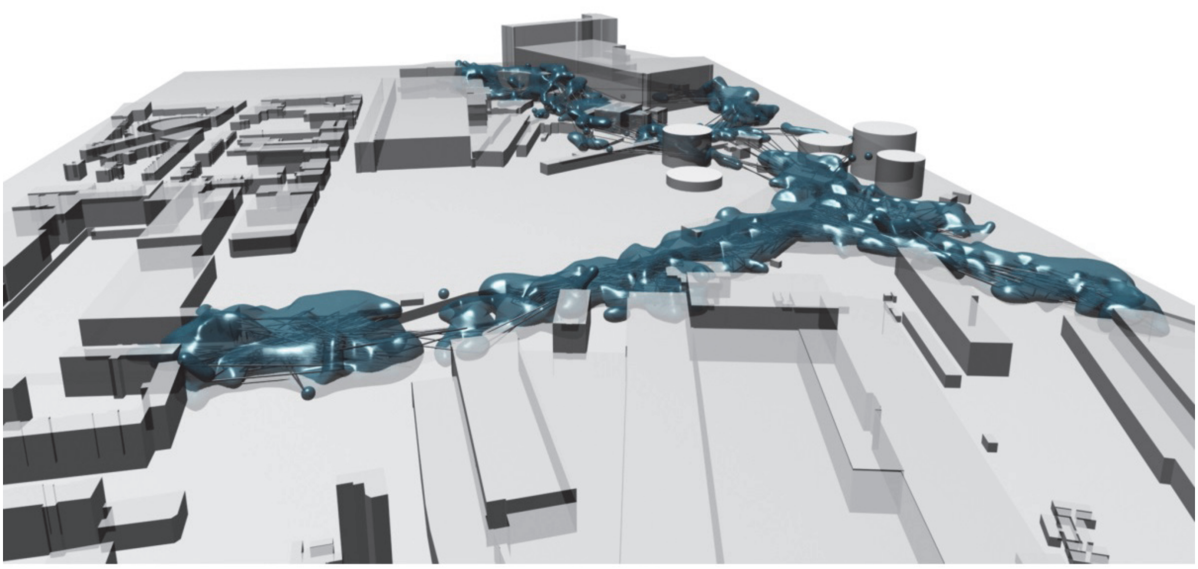


***Fig210**. SOS, 2001: a history of the SOS across the urban site (King's Cross StPancras, London) with the isosurface rendering all states of the generations together, showing a 'route' the SOS navigated for finding input samples;*

MAPPING THE MAP

The SOS produced a large number of clustered vector spaces or network configuration states that mapped locations selected by the ANN based on its perceptual structure. While the resulting network states represented a classification of the density and topology of the self-selected input vertices at a location, there was no mechanism to comparing the states and their morphologies. In 2004, the author (Derix 2004) and with MSc student Amine Benoudjit (Benoudjit and Derix 2004) re-mapped network states into a traditional Kohonen map, reducing the dimensionality to two. To do so, the weighting of ordered network nodes (i.e. their coordinates) were combined into single input vectors, creating a vector from each state and classifying them together. To test the idea, the map was trained with simple cubic spaces and multiple nested cubes.

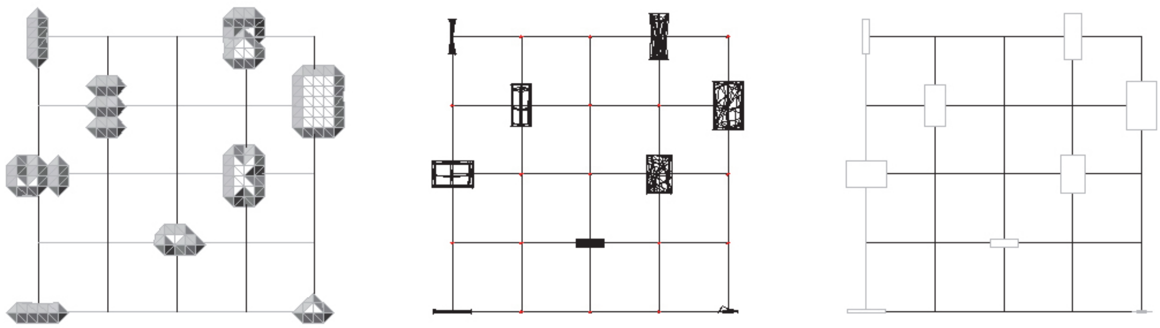


Fig211. *Mapping the SOS, 2004: an example by the author mapping the SOS network configurations into a two dimensional SOM; (right) the observer perceived input cubes varying in proportions; (middle) the SOS configurations and (left) the implicit surface morphologies; the distribution into the SOM clearly reflects the human observers' semantic interpretation of shape by adjective features*

The results of this first attempt to classify spatial morphologies were successful despite the relatively low complexity of the input samples. Classes of form could easily be identified in the map and were coloured according to their similarity within their *perceptive field* (Fig212). Where the algorithmic model would only see differences in quantities between vectors, the observer could clearly differentiate qualitative schemata such as 'wide', 'long', 'tall', 'short and long', etc. The difference in qualitatively perceived schemata is proportional to the distance on the map. And vice versa, input samples and their perceptive fields that are adjacent to each other are regarded to share features and therefore are similar (Benoudjit and Derix 2004).

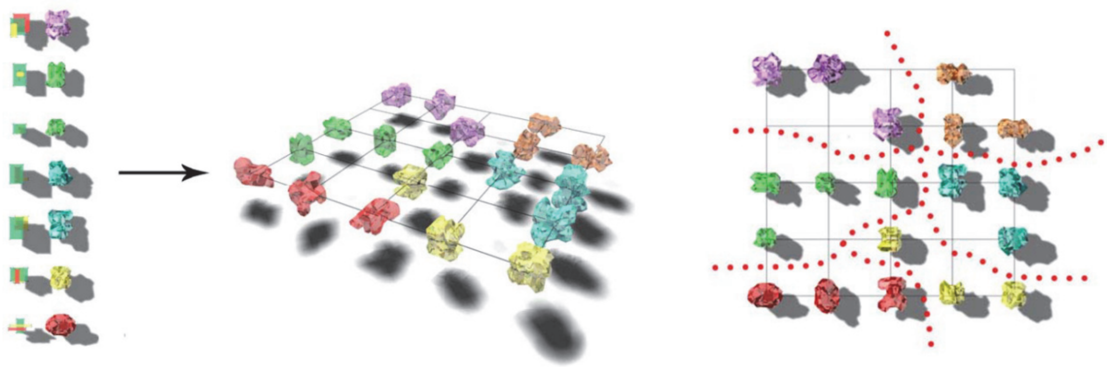


Fig212. *Perceptive Network, (Benoudjit and Derix, 2004): a slight extension of the SOS-SOM mapping with more complex multi-cubic input samples highlighting resulting clusters by colour (right)*

The measure of difference can be increased in sensitivity when normalizing the input vectors and using the *dot-product* method of comparison (Kohonen 1995, p91ff). This also allows more varied metrics to be included in a vector. The dot-product comparison does not measure each vector position individually but the normalized vectors' directions distributing the difference as a whole across the whole input sample.

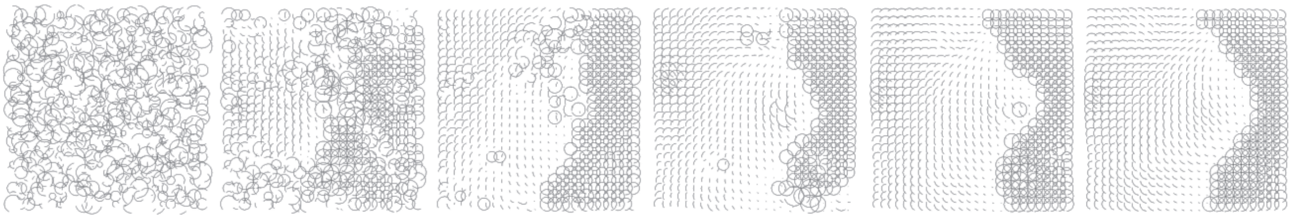


Fig213. *Dot-product SOM, 2007: a teaching example by the author showing six learning steps of a dot-product comparison of circles using a three parameter vector: opening angle, closing angle and radius; interestingly, the network revealed that closed and open circles are rationally very similar because the opening gap is equally small as the closing gap (hence the strong border)*

ISSUES AND EXTENSIONS

Three main issues arose from the SOS model based on Kohonen's SOM:

a) pre-ordering of schemata that are bounded by the input space;
b) pre-specifying the map sizes and dimensions (two, three or higher dimensionality), and the inherently fixed topology
c) re-setting learning parameters and temporal events

The provision of the urban model as a very large vertex input space, from which the SOS could select its own input sub-spaces was an attempt to decrease the interpretation of the input data, albeit all data was of the same metric. The latter experiments with cubic and other geometric input (CECA experiments included a range of diverging metrics like colour, ratios, geometric dimensions) pre-empted features for classification and often anticipated the mapping. One perceived way to solve the problem is to increase dimensionality to do pure statistical clustering but as shown in many projects such as one conducted by the author with Anna Laskari and Sean Hanna at UCL in 2007 (Laskari Hanna and Derix 2008), this approach often leads to unintelligible results where the observer is not in the position to understand the classification, because associations between data metrics are unknowable, making the mapping a 'black box'. Another approach was proposed by the author, which foresaw the autonomous collection of data by the model (Harding and Derix 2010). A prototype of a neuro-spatial robot was developed during the author's MSc that would respond to its context via sensors. A simple electro-mechanical neural robot was constructed with nodes consisting of light and position sensors driving motors, which translated activation strengths as weights to mechanical connections via gear-boxes.

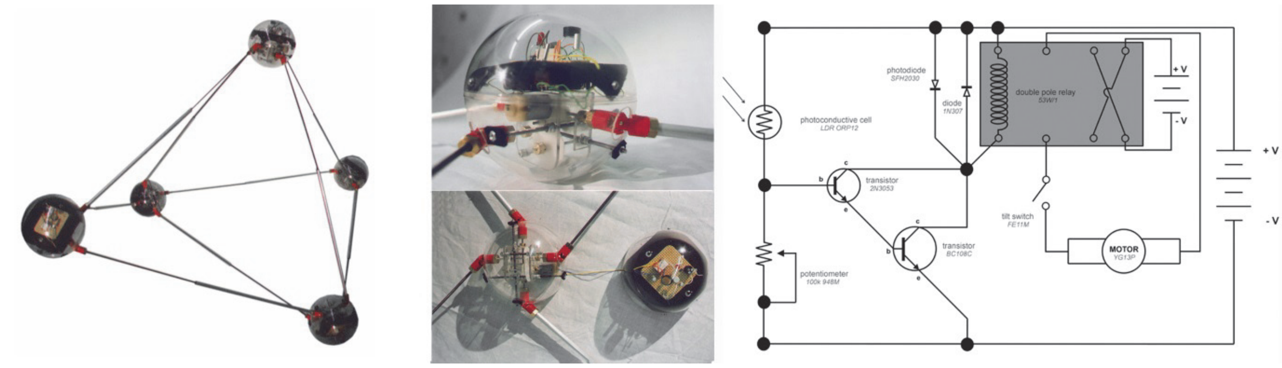


Fig214*. Neural-Space robot, 2001: (left) the double-tetrahedral space inscribed by the electro-mechanical network; (middle) a node showing the split into sensory circuit driving the motor and gear-box below; and (right) the circuit diagram for the electronics of the sensors for actuation*

The model inscribed a double-tetrahedral space and was set out on the same urban site as the SOS to physically map contextual dynamics autonomously by interacting with people. The second incomplete step of the robot prototype was meant to be an automatic transmission of the collected data to the SOS model, so that the robot collects real-world information without any pre-selection by the observer. Again any such mechanism is structurally determined by its sensor types and ranges.

6.3.3 Adaptive Topologies

Flexible map topologies and time-based adaptation was suggested by Kohonen through Growing SOMs (Kohonen 1995, p164-171) but not properly elaborated because Kohonen regarded adaptive topologies as a different epistemological problem. For temporal events such as occupation representing dynamic input data, adaptive topologies provide a good model for differentiating local network resolution and continuous learning. The author supervised CECA students Tahmina Parvin (Coates et al. 2005) and Philip Langley (Langley et al. 2007) who developed two initial adaptive topology self-organizing network for architectural application.

The basic principle for adaptive topology networks rests in the insertion and deletion of nodes and connectivity resulting from varying levels of feedback. In other words, if two nodes are not activated together for a long time, the connection is culled. Equally, a node that does not represent an input sample is culled. Inversely, new input samples occurring in input space for which no node is available, will insert a node in the network and generate a connection to the nearest (most similarly weighted) node. The basic topology for such growing networks is a *simplex*, i.e. tetrahedral map. Networks are initiated with some low number of nodes, minimally one simplex. This principle was first developed by Thomas Martinetz and Klaus Schulten (1991) to adapt a network topology to differentiated input space complexities and is based on the Kohonen SOM. Their model is called a *neural gas* network and pertains to the group of *Dynamic Cell Structures* (Martinetz and Schulten 1991). This principle was extended by Bernd Fritzke into growing cell structures called *growing neural gas* (GNG), which also fixes the learning parameters across generations for *spatio-temporal event mapping* (Fritzke 1995).

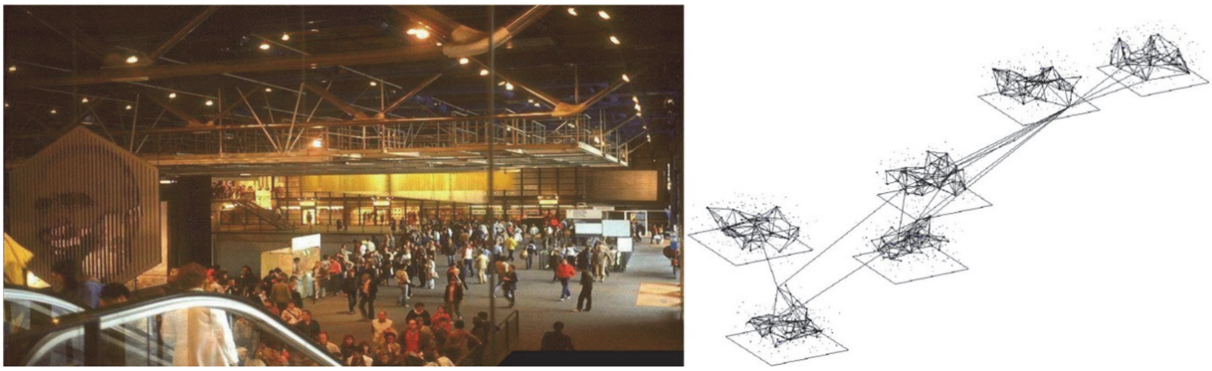


Fig215. *Event Space Neural Gas by Tahmina Parvin, 2005: (left) the interior at ground floor of the Centre Pompidou showing the open space with multiple landings that are activated concurrently; and (right) the neural gas model mapping this tempo-spatial input data on a reduced model of the Centre Pompidou (Coates et al. 2005)*

Parvin used Martinetz and Schulten's neural gas algorithm to map occupation of the Centre Pompidou in Paris[58] where several open spaces are concurrently in use. The network results in showing a differentiated topology across multiple spaces (Coates et al. 2005). Langley on the other hand used Fritzke's growing neural gas algorithm to map the distribution of human activity over 24 hours along Kingsland Road in north London (Langley et al. 2007). Langley's maps also used network analysis measures from sociology such as *cliques*, *flow* and *borders* to evaluate the properties of the topology to define territories rather than spaces. Flows were defined through *in-/ out-flow* of information, which relates to the directionality of a connection. A winning node that feeds back onto its topological neighbourhood is considered to pass information back towards its neighbours. Each connection therefore is mono-directional unless two winning nodes feed back onto each other, in which case the connection is bi-directional. The neighbourhood and flow definitions provide insights about the distribution of information and hierarchy and are pertinent to social and territorial properties. Flow integration and directionality produce boundaries within networks, which are also built on *cliques*, where spheres of influence of nodes are measured (also called *eigenvalue*). Langley visualized the connections through directed cones to reveal directionalities, cliques and borders on Kingsland Road on an hourly basis over a day.

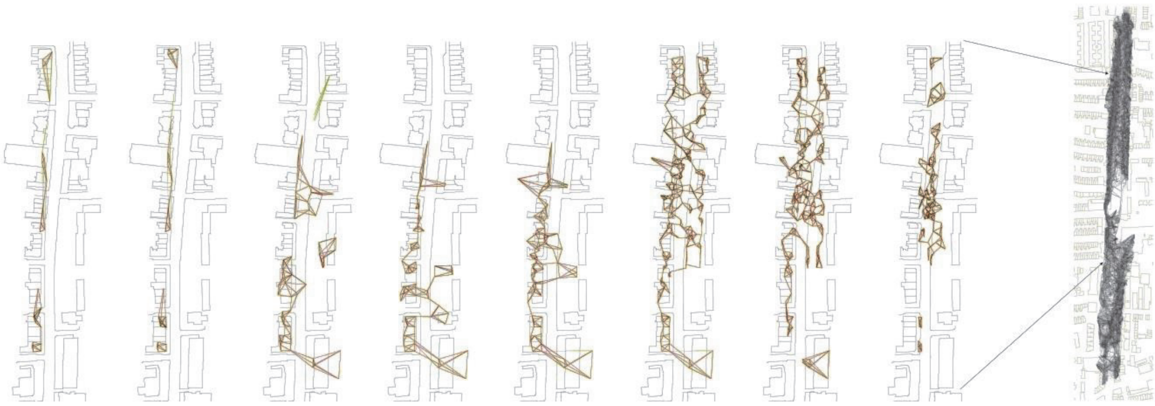


Fig216. *Territorial Growing Neural Network, Phil Langley, 2007: series showing eight hour classifications of the activation along Kingsland road in north London and (right) a summary of activation across 24 hours; the directionality of the connections can be seen in the left series through the conic rendering (Langley et al. 2007)*

[58] http://www.centrepompidou.fr

Advantages of adaptive networks are obvious when it comes to precision of highly fluctuating densities in input data samples and time-based data sets. With a growing (and shrinking) structure they manage to reflect patterns over time and scale better than SOMs. They are well suited for the visualization of time-place-activity correspondences as approximated by the walking maps at the beginning of this section. On the other hand, they lose the ability to generalize from large data sets across a wide range of metrics as already speculated by Kohonen, because they are specialized in input sub-spaces, i.e. learning selections of the whole possible set. This attributes to them a subsequent role to the general SOM where local pattern are sought (see 7.3.3).

6.3.4 Spatial Classification by Associated Qualities

To use associative networks outside of an academic context requires a generalization of the map and input format. The developments of the SOM-based models discussed started with an attempt to abolish the hierarchy from ontology so that the artificial classifier could produce autonomous schemata. By testing more complex input samples, levels of abstractions were increased and pre-defined schemata reintroduced. If associative network are meant to be applied to architectural configuration and layout problems the same questions arise as for maps and graphs: what are the minimal elements to compare and produce ratios? For the architectural profession, this is an important question as performances and efficiencies need to be comparable between buildings or designs. If the abstraction of the minimal geometric or numeric element is too low (i.e. not abstract or small enough), it is difficult to compare building layouts as the abstraction would be design-specific.

Two projects are briefly discussed that show the introduction to practice of associative networks. The first, Integrated Associative Analysis (IAA) as part of the RIBS project, used convex partitions as elemental units. The second, Space Profiler represents the generalization of the RIBS model and reduces the elemental units to discrete cells of homogenously subdivided layout plans. The IAA therefore reflects the second order spatial aggregations of graph partitions while the more advanced model Space Profiler reflects the first order discretized local values of map positions. Therefore, two types of observer are addressed, which will be discussed in the conclusion of this section. Both models were developed by Prarthana Jagannath of CDR between 2011-13 under supervision of the author.

INTEGRATED ASSOCIATIVE ANALYSIS (IAA)

Two spatial analysis models of RIBS have been discussed: the Visible Traversal Polygon Algorithm (VTPA) and the Spatial Topology Graph (STG). Seven measures were chosen from the visibility and topology network analysis and another three measures were added from local observations, to be integrated into an associative analysis network (Derix and Jagannath 2014b). The three additional measures were based on the actual layout of the building plan, represented simply by the place graph, in the form of room connectivity and depth to entrance. The first two consisted of one manually and one calculated value: a) the organizational asset value as attributed by the owner-organization and b) an interface value between asset levels. The asset interface value represents the differential between levels of assets and compares private to public assets such as the CEO's office (high asset value) and the lobby (low asset value). The bigger the differential, the higher the value and vice versa, identifying interfaces across a building plan where big thresholds between private and public functions exists that are pertinent to security planning (for complete project overview see 8.3).

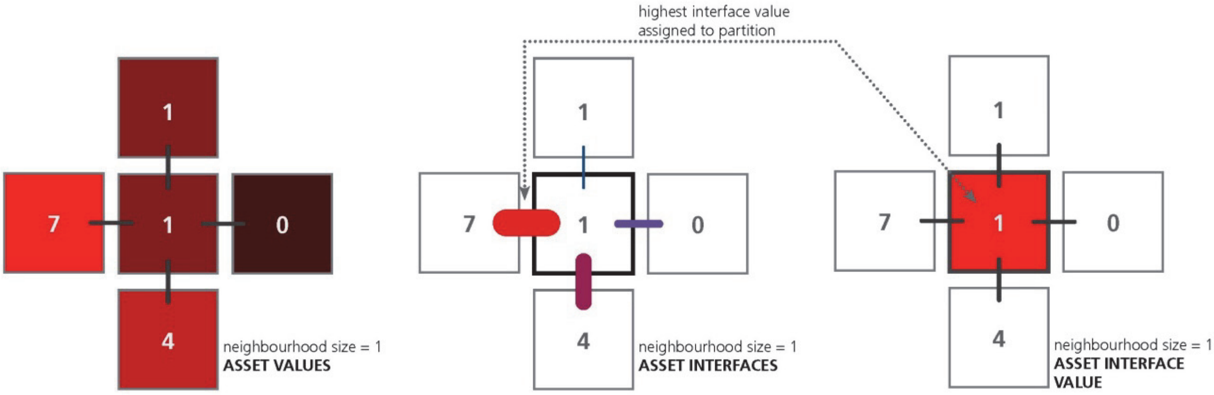


Fig217. *Asset Interface calculation for the Integrated Associative Analysis, RIBS, 2013*

Ten spatial measures resulted for each partition in a building layout plan:

<u>Asset analysis</u>

1. Maximum asset value in partition
2. Asset interface value in partition

<u>Topological infrastructure analysis</u>

3. Topological degree value of partition in graph
4. Topological closeness centrality of partition in graph
5. Topological betweenness centrality of partition in graph

<u>Visibility analysis</u>

6. Minimum visual connectivity in partition
7. Maximum visual connectivity in partition
8. Maximum visual drift value in partition
9. Maximum permeable edges in partition

<u>Additional analysis</u>

10. Proximity to entrance

The IAA aimed at providing support to security planners or architectural designers to monitor risk properties within a building and to adapt space utilization in case of security threats and general operations. All analysis models of RIBS were developed for the banking sector on an undisclosed live primary case study and unused public secondary case studies for generalization (RIBS 2013).

Node ID	X	Y	Z	Valence	Mean Distance	Circle Radius	Inbetweenness	Visual Connectivity	Permeable Edges	Drift
N0	-391	2507	449	1	54	35	899	147082	3	383
N1	-373	1835	449	1	52	109	899	1280737	9	627
N2	-373	2067	449	1	51	109	899	2064754	12	1131
N3	-368	1440	449	1	55	66	899	305046	4	473
N4	-320	1899	1975	1	59	130	899	293467	1	118
N5	-316	2355	449	1	53	109	899	857287	6	452
N6	-287	2377	1003	1	54	162	899	982719	7	909
N7	-285	1517	1500	1	56	181	899	633022	2	257
N8	-283	1403	1003	1	54	109	899	602388	3	573
N9	-281	2370	1975	1	59	169	899	311687	2	76
N10	-267	1536	-30	1	58	174	899	479297	2	201
N11	-248	2275	1500	1	54	201	899	438984	2	76
N12	-240	2356	-30	1	60	201	899	847858	2	270
N13	-235	1977	1003	3	50	212	2689	2523383	12	1653
N14	-225	2033	449	4	49	214	3581	2274836	14	994
N15	-164	1449	1975	1	58	122	899	463626	3	208
N16	-118	1814	1003	1	52	91	899	549214	2	242
N17	-98	1851	1975	3	54	83	2689	2231608	20	899
N18	-73	1545	449	3	51	123	2689	1087944	15	439
N19	-66	2168	1003	1	52	38	899	426330	3	266
N20	-57	1457	449	1	52	110	899	791029	11	478
N21	-49	1982	-30	3	55	213	2689	906722	2	154
N22	-46	1594	449	3	50	115	3575	1158277	15	377
N23	-45	2459	449	3	50	67	2689	284022	2	98

***Fig218**. Aggregate partition value matrix, RIBS, 2013: each partition consists of a series of values that can be compared against all partitions to generate classifications (here without the asset and access values)*

The partition format was decided amongst the collaborating work-package partners. A CDR proposal to utilize convex partitions of STG was rejected in favour of traditional room polygons like place graphs, which was seen as beneficial because non-architectural project partners could utilize this common representation. For the IAA, each room partition was attributed an aggregate set of values from the spatial analysis measures. Discrete position values were summed up and averaged for each partition and aggregate values such as the STG network nodal values were attributed to the nearest partition node. An aggregate value resulted from all ten measures for each partition that could now be queried for risk or just generally spatial associations.

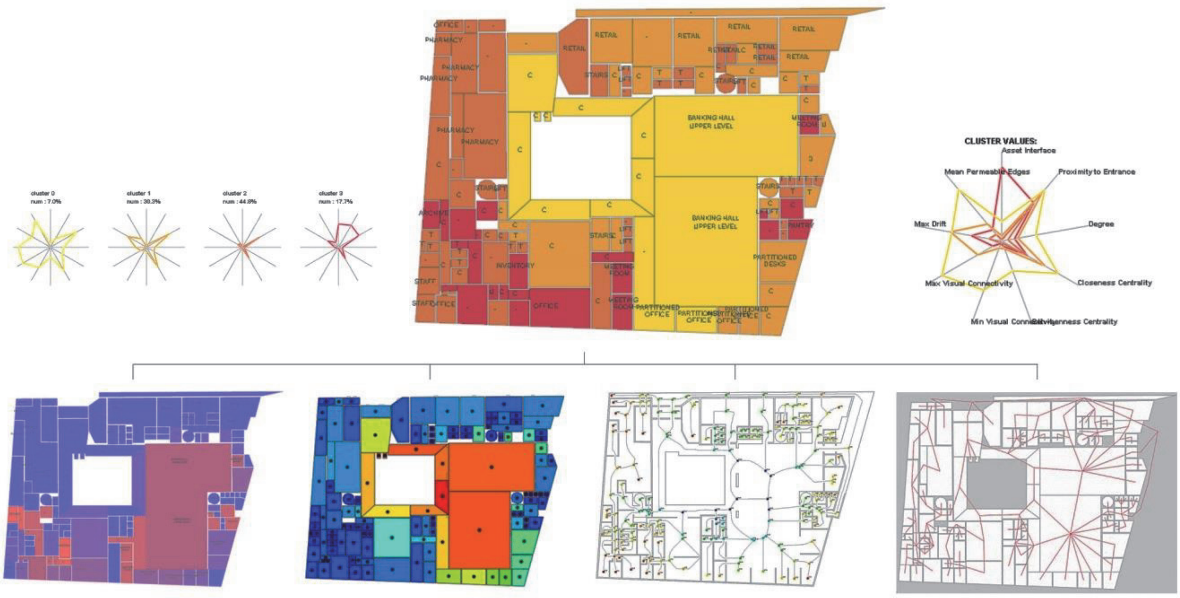


***Fig219**. Integrated Associative Analysis, RIBS, 2013: structure diagram of (bottom row) analysis models (assets, VTPA on partitions, STG and the place graph); (top middle) a visualized classification of four room clusters with (top left) the four distinct cluster spidergrams according to value and (top right) all clusters on the measure spidergram*

The associative network used for the IAA is a derivative one dimensional SOM (Derix and Jagannath 2014b). A one dimensional SOM distributes the input samples with n-dimensionality across a linear range from a maximum = 1 to minimum = 0 gradient. This range is divided into a meaningful number of clusters that provide sufficient differentiation between the weighting of the input samples. Based on testing, the efficient number of clusters was four or five. The mapping was reduced to one dimension to avoid high redundancy within the output map, that occurred when testing a two dimensional map. Lower redundancy is good when clustering discrete samples in order to achieve a tighter distribution of samples, which is the contrary for permutation maps as shown in the last dot-product map example.

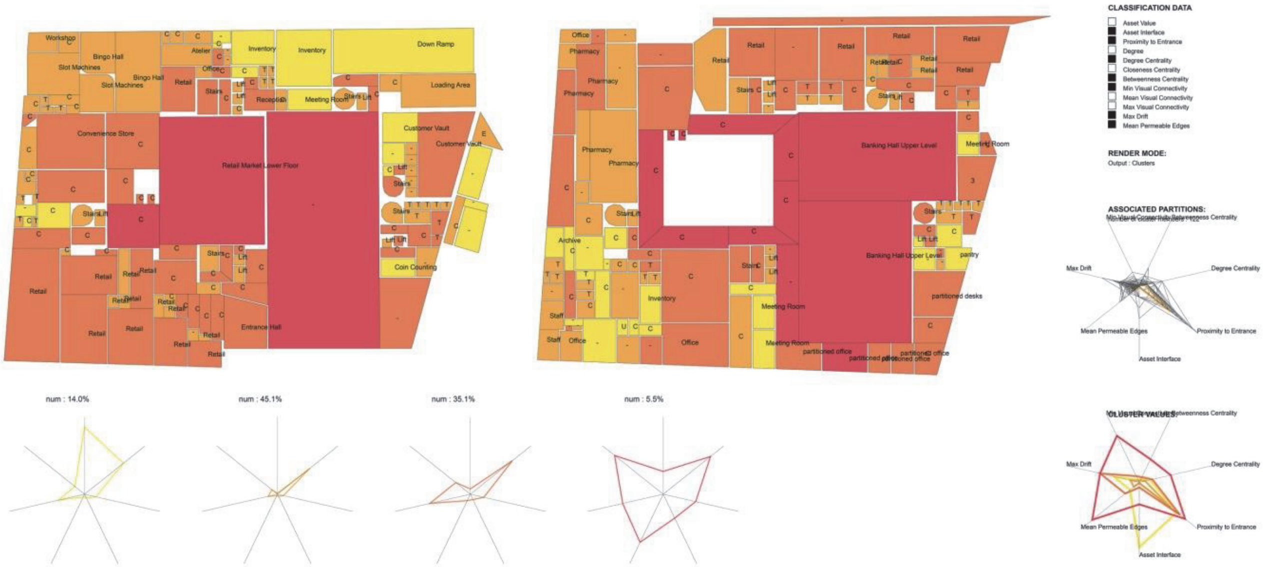


Fig220*. IAA GUI, RIBS, 2013: (top right and middle) all partitions of the layout with their cluster distribution by colour; (bottom left) the four cluster average values per measure on spidergrams; (bottom right) the average values mapped into one spidergrams for all clusters; (right middle) showing the actual values for a single partition and (top right) the selection menu for the available measures (note that the colours of this clustering map is reversed from the previous images as the colours do not consistently correspond to the cluster number)*

The IAA loads a data table containing spatial measures and the user can select a set of measures by which to classify all partitions. If a single measure like 'entrance distance' is chosen, the mapping produces only one cluster that in this case reflects the Proximity to Entrance values for each partition. Only if more than one measure is selected will a classification be calculated and clusters result from the ratios between values. Each cluster is composed of an average of its values for each measure. This is visualized within the GUI via spidergrams showing the average per cluster and the individual values for a selected partition via a separate spidergram (Fig220). The colour range of the four clusters ranges from red to yellow and is reset for each re-training, meaning that the colours vary for each mapping and are not fixed to cluster numbers.

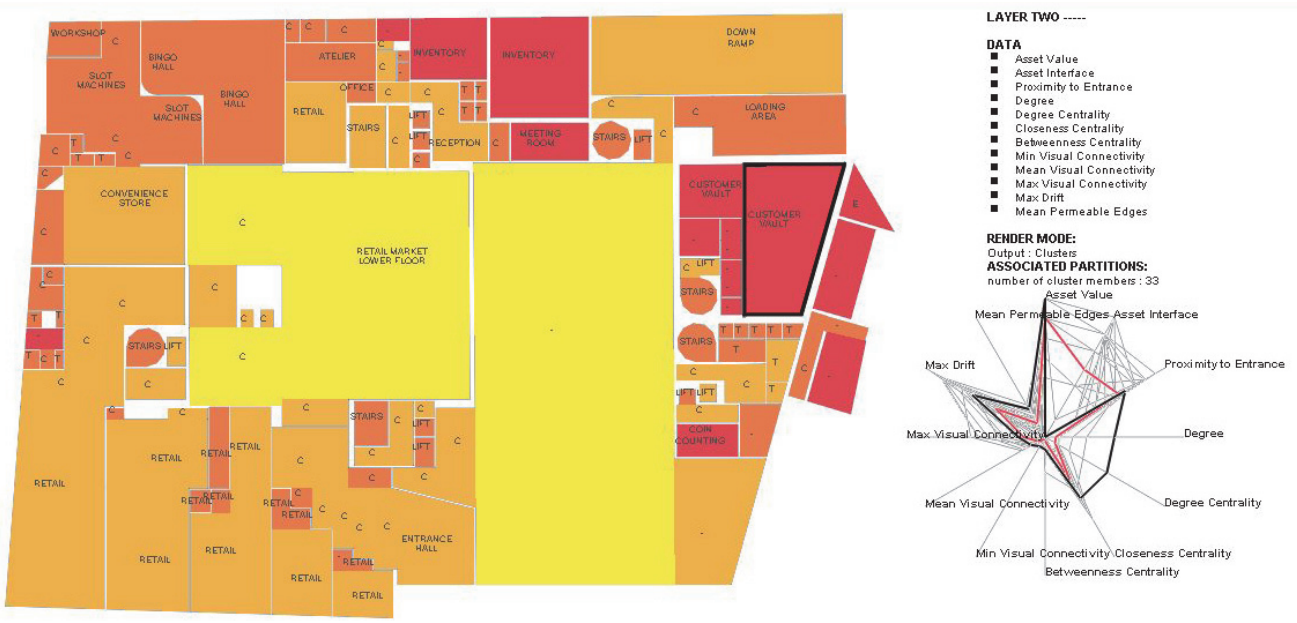


***Fig221**. IAA , RIBS, 2013: selecting a single partition like the 'customer vault' (which belongs to cluster 3 (red) and shares similar spatial properties as 12.2% of all partitions), highlights the make-up of measures in the spidergram on the right: it has high asset value and is relatively close to the entrance; but as a high risk partition, it has very low betweenness centrality, no visual connectivities and average topological connections; this is generally a good mix of qualities for a high risk asset as it cannot be 'seen' yet accessed from outside while not having building-internal public interfaces*

The one-dimensional SOM as a main interface of the IAA was complemented by two classification methods in separate interface windows: a two dimensional Mnemonic SOM to visualize the interfacing of clusters and a three-dimensional topological graph to visualize the allocation of partitions within the permeability network.

A Mnemonic SOM allows a regional distinction of perceptive fields by mapping the clusters into non-orthogonal shapes, preferably concave outlines (Mayer et al. 2005). Concavely shaped maps provide distinct allocation of clusters in segregated sub-areas of a shape, facilitating quasi-mnemonic re-identification of cluster locations when re-training the network. To use a concave map, connections between nodes across the shape perimeter are clipped, creating non-uniform topologies between nodes. Clipping the connections produces greater topological distances across the map and activation areas become very distinct. Partitions from the layout are assigned to the map nodes allowing for unassigned nodes between classes that reveal borders, occasionally producing fragmented looking maps. A pentagon shape was used to test the mnemonic SOM but also an orthogonal map could be applied for more generic interpretation (Fig222). Apart from mapping each partition into a topological relationship, the mnemonic SOM can also be toggled to visualize the asset value for each partition and the distribution of each nodal weight.

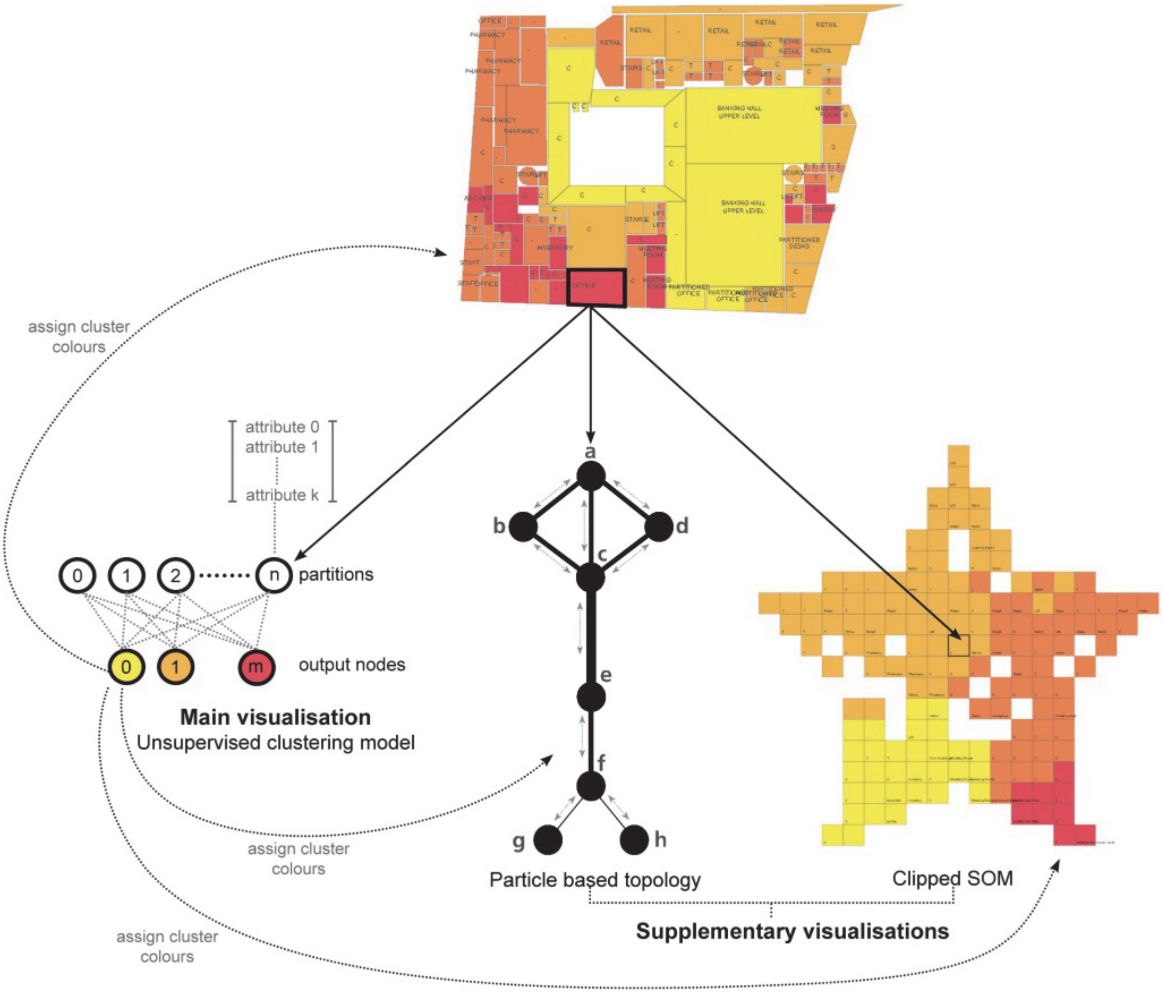


***Fig222**. IAA , RIBS, 2013: the IAA general system diagram showing (left) the one-dimensional SOM, which attributes cluster values to the layout partitions (top); a mnemonic SOM generates a classification map (bottom right) and a force-directed topology graph visualizes the permeability structure on which the nodes lay (bottom middle) (Derix and Jagannath 2014b)*

A third visualization was developed for the IAA revealing the permeability structure to the observer as a three-dimensional network, because neither the place graph nor the layout plan itself shows the intrinsic topology. Each partition is modelled as a particle in a particle-spring system, called a force-directed topology graph (Derix and Jagannath 2014b). A pair of particles is connected by a spring if the partitions they represent in the layout are one topological step away from each other in the place graph, representing accessibility between two locations. All the springs are given equal rest lengths and all the particles have mutual repulsion to each other similar to a force-directed graph (Eades 1984). The system resolves dynamically to reveal a topology of the layout, stretching out those springs that experience the most opposing forces. Springs were rendered on the basis of their topological connectivity and surprisingly coincide with the level of centrality of connections in the layout plan.

As opposed to Hillier's planar justified graphs, the particle-spring topology was developed in three dimensions for multi-floor building layouts to make the visualization more intuitive. Furthermore, the resulting three-dimensional networks approximated medial axis representations and implicitly visualized the betweenness centrality of partitions. This visualization is powerful as it highlights the interfaces between different levels of assets and associative classes on a permeability network and centrality criteria. The effects of risk scenarios like culling connections and cutting of partitions were visually made more accessible and therefore integrated well into the monitoring or design process.

Fig223. *IAA, RIBS, 2013: all analysis modes of a complex multi-floor layout such as here the Laenspar Banken, Falun. The left column shows the 1D SOM rendering the partitions (top) by asset values and (bottom) by all 10 spatial measures. The partition 'customer vault' is selected in the 'red' cluster, which contains 12.2% of all rooms, i.e. 12.2% of all partitions have a similar quality; the middle column shows the 2D Mnemonic SOM distribution of partitions with (top) the asset values rendered and (bottom) the cluster numbers rendered; the 'customer vault' partition is in the centre of the 'red cluster', showing that it does not interface with any other cluster partition; the right column shows the 3D permeability topology, clearly visualizing the circulation cycles and the outliers as according to Hillier's space types; also visible are the more 'risky' or high value assets are deeper and away from the main circulation cycles and the 'customer vault' partitions are set-off from the main circulation little accessible*

SPACE PROFILER

A post-RIBS generalization was conducted to open the associative classifier to general architectural projects. A commercial project was used as a pilot in 2013 to apply the spatial performance measures identified during RIBS. A two storey shopping mall in Lahore, Pakistan called Packages with over 100 units provided the layout geometry and 3D model. Spatial measures identified during RIBS appeared even more pertinent to the retail sector as accessibility both by foot and visually combined in its various forms provide most performance criteria such as footfall, access time, visual exposure of shop fronts, circulation structure resilience and others. Clearly, the purpose of those measures differ between building typologies such as workplace (RIBS) or retail (Packages) although their KPI are similar. For example, workplace design demands face-to-face personal interaction while retail plans to avoid direct encounters. Equally, way-finding or visual exposure is differently weighted in workplace and retail design. As opposed to a bank branch or some other functional building types, retail also incorporates large areas of semi-public circulation spaces. It is mainly the profile of this generic space which gives value to all adjacent functionally defined retail spaces. In essence, retail design – be that an enclosed mall or a high street – behaves similar to urban design and is mainly based on connectivity, accessibility and visual exposure[59].

[59] For example, the Retail Design Manual of Ireland, accessed 15.11.2014: http://www.environ.ie/en/Publications/DevelopmentandHousing/Planning/FileDownLoad,30028,en.pdf

The following spatial analysis measures are included:

Visibility Measures

1. Visual Connectivity (Integration/ Exposure)
2. Visual Choice
3. Drift (distance away from position)
4. Shopfront exposure in 3D
5. Visual exposure of routes to shops

Accessibility Measures

6. Shortest access routes (distance and time)
7. Footfall

Place Measures

8. Node degree
9. Closeness centrality
10. Betweenness centrality
11. Circulation cycles
12. Area

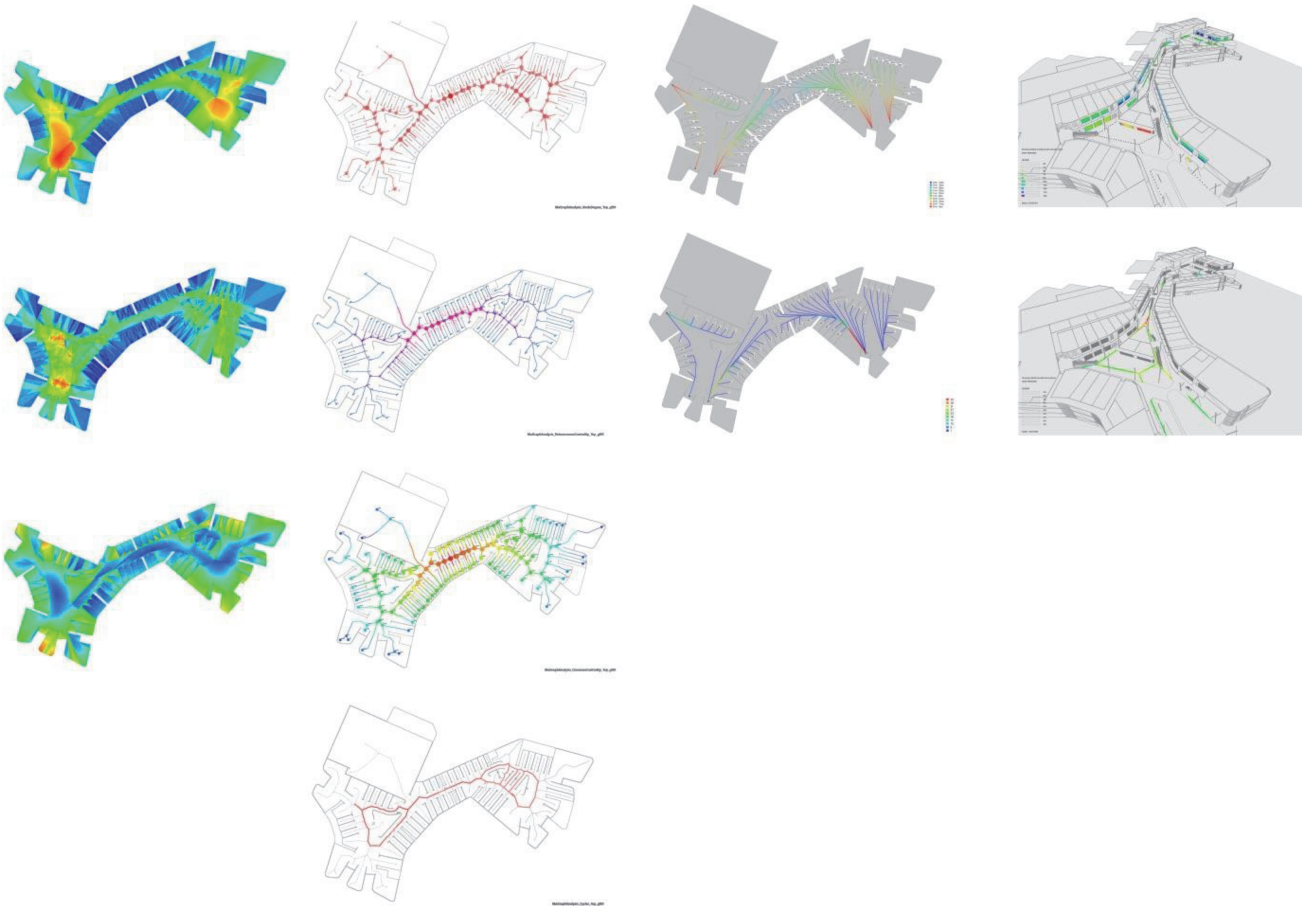


***Fig224**. Space Profiler, 2013: input spatial analysis measures from the Packages pilot, only ground floor, showing (left column) visibility measures, (second column) place measures, (third column) access measures and (right column) 3D visual exposure measures*

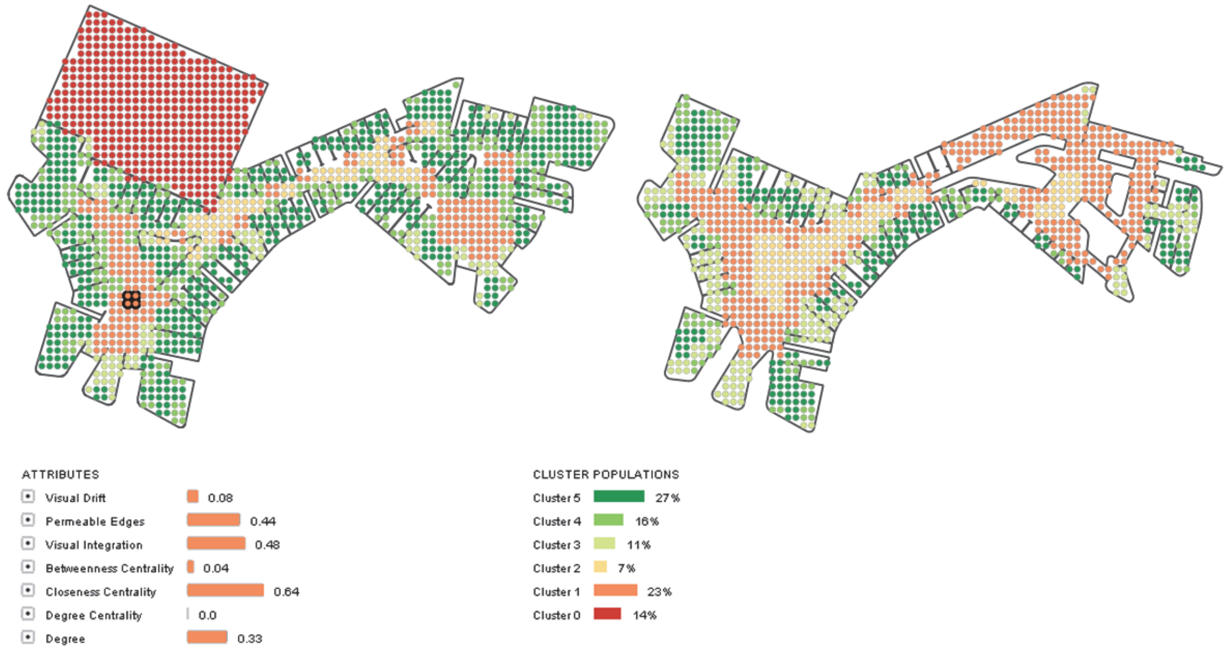


***Fig225**. Space Profiler, 2013: first version of generalization into discrete positions on the Packages two-storey layout, showing clusters in colour for locations with similar measure associations; the user here has selected one position for reading out its value composition*

The classification model uses the one-dimensional SOM of the IAA, mapping spatial measures onto partitions. Where the IAA used manually drawn partitions, the SpaceProfiler automatically discretizes the input geometry of the layout into cell positions of variable sizes that align well with the human scale or personal space (see 'proxemics' in 6.2). As the values from various spatial analysis measures are produced on different partitions that might not correspond to node positions, a diffusion of values to the positions lattice takes place. Two types of partitions are available for evaluation that the designer-user can select from: discrete spatial positions or area schedule unit partitions.

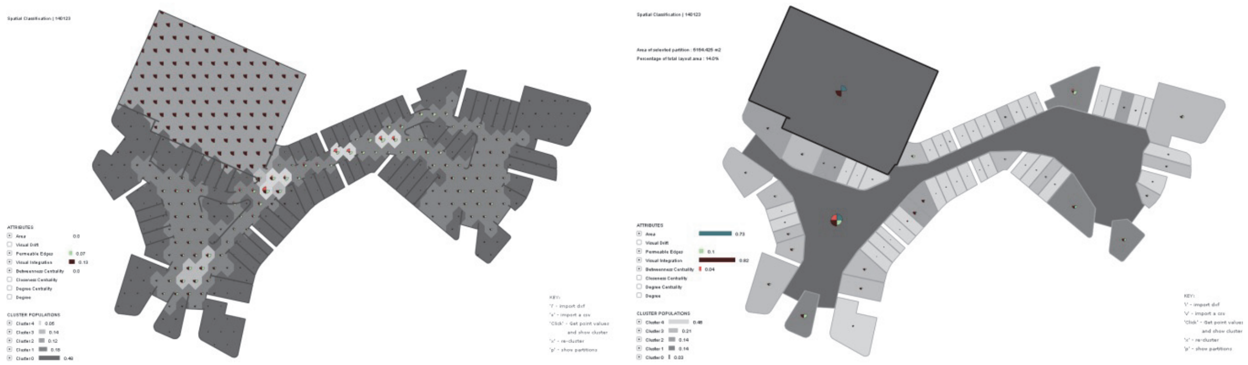
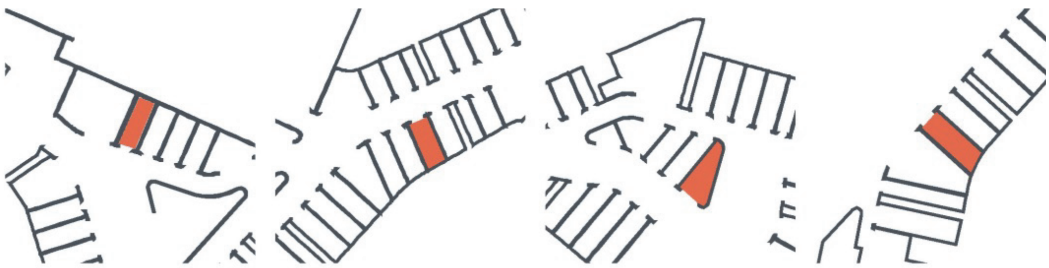


***Fig226**. Space Profiler, 2013: second version showing (left) the position based clusters with overlaid chart visualization and (right) the partition based clusters with overlaid visualization for some selection of analysis measures*

A series of graphic tests was conducted for a more generic visualization and a dual rendering method was developed that separates the cluster rendering from the value composition rendering at each position/ partition (Derix and Jagannath 2014b). The clusters are rendered by grey-scales and value compositions are rendered in colour and scale as pie-charts for each position or centre of partition, based on Charles Minard's 'meat catchment' visualization, which shows which French regions contribute to Parisian butchers how much and what type of meat (Tufte 2006, p131).

As with any analysis model, results require interpretation by observers. The Space Profiler does not deliver automatic reporting and resulting analytical maps require interpretation for which both the visual maps and the exportable exported data tables with values per position/partition by selected measures are available. The use of a Space Profiler might not be restricted to architecture but in fact aims at strategic decisions made after the concept design stage. The decisions in this case are not to be made by architects but by clients such as developers and their consultants like investment consultants who anticipate the unit mix and allocation within a proposed development. Depending on place classes, tenant profiles are established or vice versa, tenant profiles are attributed to available place classes.



	RETAIL REFERENCE #1	RETAIL REFERENCE #2	RETAIL REFERENCE #3	RETAIL REFERENCE #4
WALKING DISTANCES	86m / 1 min	202m / 2 min 24 sec	75m / 53 sec	121m / 1 min 21 sec
VISUAL DECISION POINT	medium to high *(yellow to orange)*	low *(blue)*	low *(blue)*	medium *(green)*
VISUAL EXPOSURE 2D	medium *(yellow)*	low to medium *(light blue)*	high *(orange to red)*	medium *(green)*
SHOP FRONT EXPOSURE	low to medium *(light green)*	low *(blue)*	low to medium *(green to blue)*	low to medium *(green to blue)*
LOCATION CENTRALITY	low to medium *(purple)*	High *(red)*	low to medium *(purple)*	medium to high *(pink)*
CIRCULATION LOOPS	1 step away from a loop	1 step away from a loop	1 step away from a loop	1 step away from a loop
CONCLUSIONS	**Moderate** This shop unit is fairly close to the entrance, at an important decision point and is in a highly exposed area. The shop front however has low-mid exposure and the location is not very central to the layout. Even though this indicates a low footfall in front of the shop, footfall may be improved due to its proximity to the entrance.	**Poor** This shop unit is farther than any other shop unit to the entrance. It is a low value decision point and is poorly supervised. It also has low shop frontage. The location is however very central to the layout and may incur relatively higher footfall for this reason.	**Good** This shop unit is close to the entrance and is in a well supervised area of the layout. It is however a low value decision point and is not very central to the layout. Since the closest entrance to this unit is the entrance with the highest flow / footfall (see pg 3), it is likely to incur heavy footfall.	**Poor to moderate** This shop unit is fairly far from the entrance and is in a moderately well supervised area of the layout. It is also at a moderate decision point and is highly central to the layout. Even though the shop frontage is low other factors may induce a high footfall past the unit.

Fig227. *Space Profiler, 2013: extract from project report to client (Aedas), showing four sample locations for pilot evaluation from six measures; the evaluation is purely done on the analysis measures, not on any extra commercial KPIs (Aedas|R&D 2013)*

6.4 CONCLUSIONS

Three types of associative networks have been discussed: fixed topology spatial networks (3D SOM), fixed topology maps (1D and 2D SOMs, dot-product SOM and Mnemonic SOM) and adaptive topology networks (neural gas and growing neural gas/network). While the purpose of development has always been an autonomous cognitive mechanism to reveal patterns of association, their use has been transformed from academic research into spatial organization for professional development of layout profiling. For this to happen, an adjustment in the representation of the spatial unit has occurred that utilizes both discretized positions as in 5.2.1 *Maps* and aggregate partitions as in 5.2.2 *Graphs*. There appears to be a correlation between the abstraction of the spatial unit and domain association. First-order abstractions such as positions allow a more general comparison of places within global configurations while second-order abstractions such as partitions provide comparisons between detailed specifications of place.

Compared to the two previous sections of this chapter, it is almost meaningless to try to summarize correlations between network types, their behaviours and relevance to practical applications into a matrix. Just for the sake of completeness, the below table is provided limiting itself to previously introduced categories. As any dimension can be introduced to the data sets, any kind of association could emerge, making the table a small extract of the possible association space.

MEASURES		SPATIAL MEASURES							VISUAL MEASURES						TOPOLOGICAL MEASURES							
ASSOCIATIVE NETWORK	COGNITIVE AFFORDANCE	interestingness	inside/outside	next to	narrow/ tall/ big/ small etc	time	co-presence	spatial interfaces	visual integration	openness	occlusion	drift	spatial exposure	route exposure	borders	cliques	directionality	depth	degree	betweenness centrality	closeness centrality	cycles
FIXED TOPOLOGY SPATIAL																						
three dimensional	morphological patterns	X	X	X	X										X	X						
FIXED TOPOLOGY MAPS																						
two dimensional	geometry cognition		X	X	X			X														
two dimensional	data association	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
two dimensional	configuration permutation search	X					X								X							
one dimensional	configuration weighting						X	X	X					X						X	X	X
one dimensional	place profiling							X	X	X	X	X	X		X	X		X	X			
ADAPTIVE TOPOLOGY																						
Neural Gas	event-space density patterns					X		X							X							
Growing Neural Network/Gas	place-activation association		X			X		X							X	X						
Growing Neural Network/Gas	social inclusion profiling		X			X		X							X	X	X					

Fig228. *Association network model to cognitive affordance and measures matrix: this table is purely for the sake of completeness of chapter six; in practice and theory, any kind of association can be generated from available input data as long as the data is comparable and hence often needs normalizing to be compiled into a vector*

6.4.1 Unit Abstraction

In section 2.7.3 a quote by Hillier and Leaman (1974) was used to support the New Epistemologists' argument that *designing* refers to implicit knowledge of two kinds: the experience from 'being in space', i.e. the observer as user generating empirical knowledge from occupying, using and participating in social norms of use; and as designer with heuristics of designing by learned procedures that correlate empirical knowledge. The frame of the quote is very relevant in the context of autonomous associative networks and reads: *"The key to understanding how such structures ('formal structures') combine into effective artificial systems must tell us about the nature of the morphological units, the building rules of the overall logic, and how the two combine together in a 'natural' way. The thesis here is that the elementary units of the morphology are not 'units' at all in the usual sense, but are already structures. Moreover, their structure, as opposed to their phenomenal form, is as autonomic as the overall formalism. The understanding of all such systems lies in discovering how the internal autonomic structure of the 'simplest structures' of the morphology already contains the rules which govern aggregation into higher logical forms. The failure of general system theory to progress beyond an elementary level in characterizing how such systems work is because this elementary principle of the* *dynamics of artificial systems cannot be formulated within a definition of a system as 'elements and their relations'. There simply are no elements."* (Hillier and Leaman 1974, p6). The quote directly supports Coyne's proposal for connectionist models for intuitive design: *"There is no explicit representation of a schema. However, a schema is implicit in the pattern of associations generated by the system during the learning process. Their model of how schemata are derived from the examples is thought to accord well with cognitive processes".* (Coyne and Newton 1990, p40).

Both suggest that spatial knowledge of the designer as observer must be addressed generically when used in an open design context without fixing relational structures such as parametric dependencies with project-specific ontologies as John Gero proposed (see 2.8.1). Self-organizing associative networks conduct their own structuring of associations and produce, within limits, their own schemata. To become an agency for a generic observer, it was shown that the elemental unit had to become so small as to be no element at all in an ontological sense but simply data features from which to build relational structures from. The more abstract and elemental the spatial units become, the less a specific project type or user is addressed. While models such as the Space Profiler can be trained for specific ontologies and revert back to expert systems if necessary, their generalized mechanism allows them to be a design reasoning platform for many other project stakeholders who work with different ontologies.

As demonstrated in the both the IAA and Space Profiler, the profile and value of a location as a partition, and to some degree position, is heavily reliant on connection spaces such as circulation. Fluctuation in profiles mainly occurs within semi-public circulation space (equivalent to Hillier's b-, c- or d-spaces) from which functional or 'end-spaces' (or Hillier's a-spaces) derive their value. This connectivity space represents configuration and generic functions of occupation but as discussed before, has little specification and is therefore under-constrained. Finding a good elemental spatial unit definition for such under-constrained spaces is equally difficult as shown in all graph and network models discussed so far. The transition from the IAA – which uses the place graph area representation into simple 'rooms' – to the Space Profiler – which uses map-like discrete cell positions – revealed that semi-public space where transitions and interfaces occur are not well represented by either definition of aggregate partition or discrete positions. The structural skeleton approximates a natural partitioning of a permeability structure through the configuration of its edges but does not represent many other properties. The discrete positions render the quality of public space with its interfaces better than partitions but in practice that has so far not been of interest where explicit values need attributing to explicit boundaries. This points toward multi-scale representations of partitions as attempted in the Space Profiler where different KPIs can be encoded via appropriate visualization. The developments in this section have improved upon the representations for spatial analysis given by Space Syntax, particularly in the combination of computable partitions with *designerly* interactions on analysis measures. Models such as the IAA and Space Profiler implicitly point towards typologies of qualitative configurations classified by user and occupation-centric measures. But they have not quite resolved the task that Philip Steadman outlined in 1983: *"[…] rooms are set along relatively simple and coherent circulations* *systems consisting of a few branching corridors which extend along the buildings' whole length. There are many dissections which are made up, by contrast, of a deep maze like agglomeration of overlapping rectangles, many of them completely internal and through which any linking pattern of circulation routes would be circuitous and confusing. If we could capture properties like these in explicit geometrical measures, then we might be able to limit the study of dissections, for example, to a much reduced class of arrangements which would all be 'building-like' in some well-defined sense.*" (Steadman 1983, p171)

6.4.2 Representational Fidelity of Behavioural Assumptions

A transition from continuous perceptual to discrete perceptual fields has taken place. Continuous perception was mapped via agents whereas discrete perception through calculations on discretized territories represented through meshes, graphs and networks. For two reasons graph representations of typical behaviours are currently replacing many applications of agent-based representations when *designing*: a) focus on general occupants' behaviours and b) speed and scale of application.

It is tempting to over-specify agent-based models in an attempt to integrate as much complex and individualistic behaviour as possible. But there are many problems with this approach as outlined by Andrew Crooks (2008) or Brian Epstein (2011). Epstein challenges the anthropocentric fallacy which in agent-based models assumes that individualistic behaviours are determined by cultural or social properties. As Crooks (2008) also states, when too many behaviours are 'aggregated' their representation of processes becomes difficult to specify and hence evaluate (see discussion on parsimonious models in 7.1.3, 8.3.2, 9.2.1 and 9.2.4) and relevant processes hard to identify. Further, in professional agent-based models with many individualistic behaviour rules, it is common to run a simulation several times to identify recurrent patterns (Arup MassMotion approach[60]). This highlights the paradox that despite the possibility of individual behaviours, the value of behavioural modelling in architectural design rests with general behaviours across an identified population and to find statistical means. Outliers of situated individual behaviour or whole model states do not provide an indication of typical occupation.

Graph theoretical representations with network analytical measures are geometric properties that correlate to general behaviours of populations or sub-groups of users such as described above. In a *designing* context, where a spatial configuration is meant to approximate occupancy affordances this generalization is more useful than individualistic exceptions. Epstein (2011) also points towards the locality fallacy where ontological 'local' specifications of the agent's programming can be confused with behaviours at geo-spatial locations.

For *designing*, speed and scale also plays a major role. Agent-based models for multiple behaviours across a population take much longer to process than graph representations. Hence, agent-based modelling is mostly conducted post-design for validation and compliance testing. Crooks (2008) also mention the necessity for

[60] http://www.oasys-software.com/products/engineering/massmotion.html, accessed 12.10.2015

calibration, verification and validation of complex agent-based models, i.e. the careful weighting of agents' rules that would be near impossible in a synthetic generative-analytical design methodology. Clearly, also graphs and networks can be weighted for individual behaviours to produce faster approximations of spatial patterns but again the assumptions for this class of individual would increase making it even harder to validate results.

Fig229. *Four instances of an access network for the Euston Crossing project (7.2.2): the network is weighted from 0% (left) to 100% (right) to approximate individualistic behaviours by routing origin points to a destination point through preferred locations such as cafes, newsagents and others; the standard 0% deviation graph represents shortest routes and the colours indicate the flow load along graph edges when preferences change*

7 SYNTHETIC CONFIGURATIONS | ASSOCIATIVE GENERATION

"Spatial elements, […], are properly seen not as free-standing 'elements', with intrinsic properties, waiting to be brought into combination with others to create complexes of such properties, but as local spatial strategies to create global configurational effects according to well-defined laws by which local moves induce global changes in spatial configurations." (Hillier 1996, p284)

Models in 5.1 discussed generative algorithms and the knowledge they unlock for spatial design. Correlations between algorithmic models and design aspects were identified that support the generation of certain spatial configurations from the perspective of combinatorial or objective performances. Models in chapter six introduced a series of algorithmic representations that analyse the configurations of spatial objects in the field for cognitive and behavioural affordances. Those provide metric instruments in the form of local values, global configuration diagrams and type classifications that help to guide the generative process for human-centric performances. Hillier (1996) proposed that combinatorial complexes are governed by spatial strategies (generative rules) according to well-defined configurational laws (empirical laws) but did not have the chance to apply this proposal beyond the *barring process* (Hillier 1996, p239-245)[61]. In this chapter, a series of models are discussed that synthesize generative algorithms providing combinatorial rules with spatial strategies that originate from human-centric performance indicators. Synthesis in this chapter is understood within the generate-and-test tradition although the testing is conducted concurrently during the generative process rather than afterwards as commonly done. The approach still distinguishes between the observer and the model by differentiating different levels of observer agencies as done in chapter five, where the observer as designer always remains an external mediator. Therefore, the observer-designer still functions as the global component of the design system with the synthetic model consisting of locally distributed components and actions. However, as opposed to directed generative algorithms with explicit target values in chapter five, the role of the observer's guidance is becoming increasingly aligned to and constrained by human-centric performances and targets are replaced by objectives. Thus, states emerge that are heuristically 'good' configurations resulting from performance monitoring instead of striving towards numerical optima, implementing Herbert Simon's concept of aspiration levels (see 3.3).

Structure

The structure of this chapter differentiates the increasing emancipation of the synthetic model as laid out in chapter five and six, starting with global configuration evaluation by design intentions of the observer and arriving at local strategies evolved by the model, generating apparently autonomous intentions through independent associations.

---

[61] The *barring process* is a unique example in Hillier's writing as it assumes a *designing* observer, i.e. some interactive iterative process, not taking a finished or an automatically generated configuration for granted. Paul Coates and the author have pointed this lack of re-integration out to Alasdair Turner, Bill Hillier and Alan Penn at UCL for many years.

Three sections describe the relationship of the observer to algorithmic components of the model and are arranged by:

1. Remote Observer: agency of observer who intrumentalizes field knowledge to constrain generative models from 'above', as it were. Design intentions are global yet locally generated as properties of the configuration.
2. Situated Observer: agency of observer through interaction at local level to nudge algorithmic processes. Performance evaluation using field knowledge and observer are mutually autonomous and their interaction takes place locally, from 'within', as it were. Global performances are therefore mediated consensuses based on a hybrid epistemology of observer and algorithmic intentions.
3. Learning Observer: field performances are situated in the learning model and in the observer input about the field. This input however is non-schematized and the model generates configurations by learning to construct performances through association. Planning intentions are provided in the form of selecting non-hierarchical input features and emerge from the generative process.

7.1 REMOTE OBSERVER | INSTRUMENTALIZING FIELD KNOWLEDGE

Object configurations based on observer compositions discussed in chapter five separate the heuristics of the generative process from the performance evaluation set globally by the observer. KPIs therefore do not correlate to the implicit logic of spatial configurations or their knowledge production process, representing explicit numerical targets based on measures of shape such as area schedules. The core algorithm discussed in chapter five for composition of geometric shapes is the genetic algorithm, which uses the fitness function to measure the performance of a phenotype against targets. As discussed in 5.4, GAs are ideal for assembling geometric shapes into complex compositions since the representation of input via chromosomes allows a hierarchical description that the embryology can en-/decode parametrically. The disjoint between fitness function and meta-heuristic process (Darwinian evolutionary process of cross-over and mutation) provides the observer with global control of the developmental process. Models in this chapter will replace global external targets with inherent performance targets based on the structure of the embryology. While the embryology itself en-/decodes the geometry of spatial configurations, the fitness function does not represent simply numerical targets but ratios calculated from the embryology measuring layout performances in terms of occupancy potential. Arbitrary numeric targets are replaced by cognitive mappings and behavioural diagrams, where natural selection decodes field conditions as KPIs.

7.1.1 Emergent Circulation

A direct performance evaluation by a GA fitness function is illustrated by a student exercise in 2007 at CECA. The brief for the student task was similar to the 2006 project brief described in 5.3: develop an architectural application for a GA with the embryology encoding spatial partitions and the fitness function evaluating an occupation potential. The author's brief specifically asked to integrate isovist analysis for the fitness function for layout evolution and was called the Wall-to-Vision. The concept itself was based on an unpublished diploma thesis by Joerg Kraemer and Jan-Oliver Kunze (2005) at TU Berlin, who combined evolutionary algorithms with VGA.

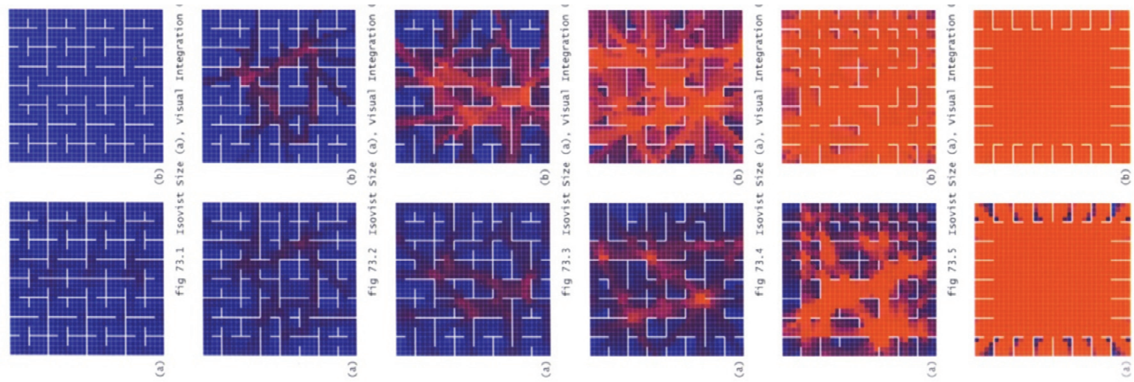


***Fig230**. Kraemer and Kunze, Design Code (2005): eroding partitions produce different levels of (bottom) isovist sizes and (top) visual integration (original rotated by 90°)*

Whereas Kraemer und Kunze evolved wall partitions reminiscent of Hillier's *barring process* in order to produce visibility conditions, students were asked to use the isovist analysis to evolve some connectivity pattern for circulation on an abstract building floorplan. Philip Langley used a simplification of the discretized grid connectivity graph (see 6.2) for the isovist analysis as a fitness function. The fitness of each individual in the population of simple partition grids consisted of a ratio between good accessibility and wall partition length:

individual.fitness = 1/ (((sum(intervisible_nodes) + sum(connection_lengths)) / sum(wall_lengths))

'Good' accessibility was measured via the sum of both number of visible grid nodes and lengths of connections, applying a bias towards long isovists akin to the *drift* measure (6.1). This double sum ratio ensures that the fitness balances performances of locally integrated positions with global accessibility. Eroding all wall partitions theoretically guarantees highest connectivity between all grid nodes but inversely cancels out any spatial differentiation into circulation and spaces. Hence, the isovist fitness is divided by the sum of wall length segments (not the number of wall segments) and is normalized, so that well-connected individuals with long sightlines and high wall lengths are rewarded.

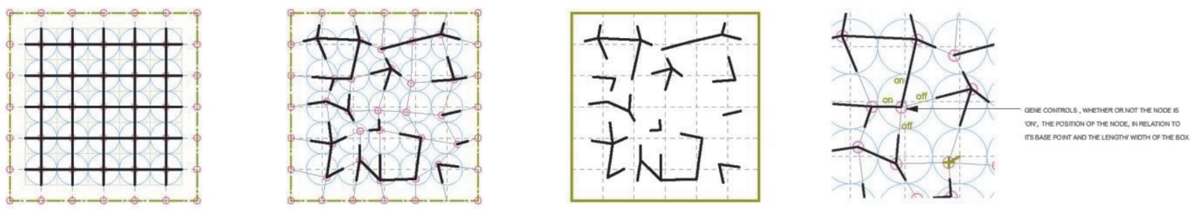


***Fig231**. Wall-2-Vision, Philip Langley (CECA), 2007: the embryology with nodal offsets and binary wall segment activation*

The embryology is based on a grid subdivided equally in x and y and each node can be offset against the (x,y) grid coordinate by a maximum radius that does not interfere with neighbouring cells. The node is connected to four wall segments that can be binarily switched on or off. Hence, each chromosome consisted of five genes specifying 1: offset of node to (x,y) grid coordinate and 2-5: four binary position specifying if a wall segment was on or off. The selection function uses the roulette wheel with two individuals in each generation selected for breeding.

Fig232. *Wall-2-Vision, Philip Langley (CECA), 2007: three randomized initial conditions (left) with varying geometric primitives evolve consistently ring-like circulation patterns with local clusters of spaces (right); the three experiments used slightly different fitness criteria, which shows in the top experiment creating larger mono-clusters*

Despite the simple fitness and selection functions, a clear structuring of the partitions could be observed and monitored numerically. Langley tested several geometric primitives as field partitions but all performed similarly: from an unstructured initial randomly seeded field of partitions a series of clusters emerged that connected via a ring-like circulation, increasing the global accessibility while ordering the field locally. This result appeared reminiscent of the *beady ring* structures that evolved in the '3 syntax' of Hillier and colleagues' (1976, p176) original *space syntax* theory, which also attempted to balance permeable and solid spaces into connectivity configurations.

7.1.2 Accessible Assemblies

In a professional setting, optimization using a meta-heuristic algorithm is not applied to generate configurations from scratch without some pre-processed ontology, unless the scale is constrained to a controlled simple problem like a furniture grid[62]. As demonstrated on the Khalifa-bin-Zayed residential tower in section 5.1, some minimal spatial unit needs to be constructed such as an apartment unit and an accommodation schedule. While the previous case study showed the application of mapped discrete positions, another model is presented where the evaluation works on a spatial aggregate (permissible apartment layouts) whose configurations are controlled by multiple inherent performance criteria:

- Accessibility from core (lifts and staircases) to apartments
- Orientation all balconies and windows are facing the external façade
- Tightness number of 'empty' cells on floor and overlaps

Fig233. *Dudley House, CDR, 2009: Aedas proposal for a 14 storey residential tower at Paddington basin, LB Westminter*

Dudley House was a request-for-proposal in 2009 by LB Westminster and Westminster Community Homes used to explore the possibility to develop a capacity automation tool. The residential complex included a 14 storey tower with a single core. A catalogue of apartment types was drawn up including 13 alternative layouts for one and two bedroom apartments, whose parameters consisted of habitable floor space, balcony, window fronts and entrance options. The task was to find efficient combinations of apartments per floor that would satisfy the accommodation schedule, providing capacity forecasts and mix of apartment types were subject to

[62] This is specific to a professional context where timelines of phases often do not allow the application of computation when architects have fixed heuristics that produce solutions much faster than calibrating a model and running it, unless a generalized phase-specific meta-heuristic model is available.

changes. The meta-heuristic search algorithm chosen for this task was a GA adapted from the Khalifa-bin-Zayed model, developed by Åsmund Izaki (Helme Derix and Izaki 2014).

The embryology consisted of chromosomes representing five integer and float number genes for

1. Apartment type from catalogue
2. Rotation of apartment (4 orientations)
3. Mirroring of apartment (2 orientations)
4. Grid position x-axis coordinate
5. Grid position y-axis coordinate

Five fitness criteria composed an individual's fitness consisted of a weighting between

1. Accessibility   number of apartment entrances in reach
2. Overlap   number of overlapping apartment cells
3. Neighbour   number of occupied neighbouring cells (tightness)
4. Balcony   balcony cells must align with perimeter and can cantilever
5. Daylight   window cells must face perimeter yet perimeter contained

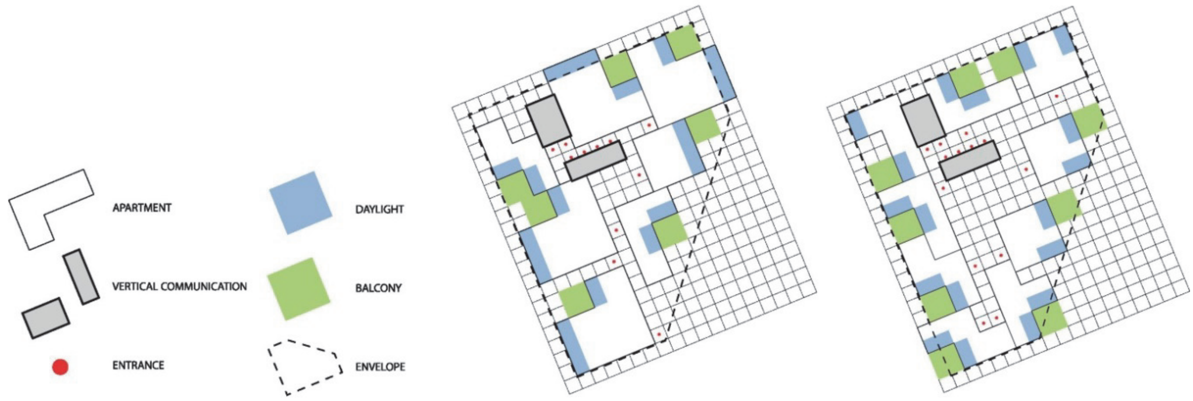


***Fig234**. Dudley House, CDR, 2009: (left) the key to the colour and polyline codes for the model and (middle) a randomized initial configuration and (right) a 'good' solution by the model for 9 x type 1 (single bedroom) with all fitness criteria resolved*

This resulted in an individual's fitness of:

*individual.fitness = sum((access * 2) + (no. overlap * 2) + (no. neighbours)*
+ exterior_balcony + (no. lit_windows))

Fitness criteria were dependent on the colour of the cell for which tests were conducted. To evaluate the accessibility between cores (several positions around the core were valid for access) and entrance cells (also several grouped positions were provided for each apartment), a floodfill algorithm was used to establish the number of apartment entrance cells reachable by core access cells. If any of the apartment cells could be reached, accessibility was valid, not guaranteeing the shortest access route. In the fitness sum, accessibility and overlap were valued twice as highly as orientation and daylight, to weight the sum. The floodfill algorithm samples the grid for cells with target values, here the apartment entrance colour value. Similar to the

People Movement II model in chapter 5.2.1 that samples territorial positions, the 'flood' diffusion acts like an agent searching for a target in a seeded field, which reflects states of territory (the cells could express any state through any colour gradient).

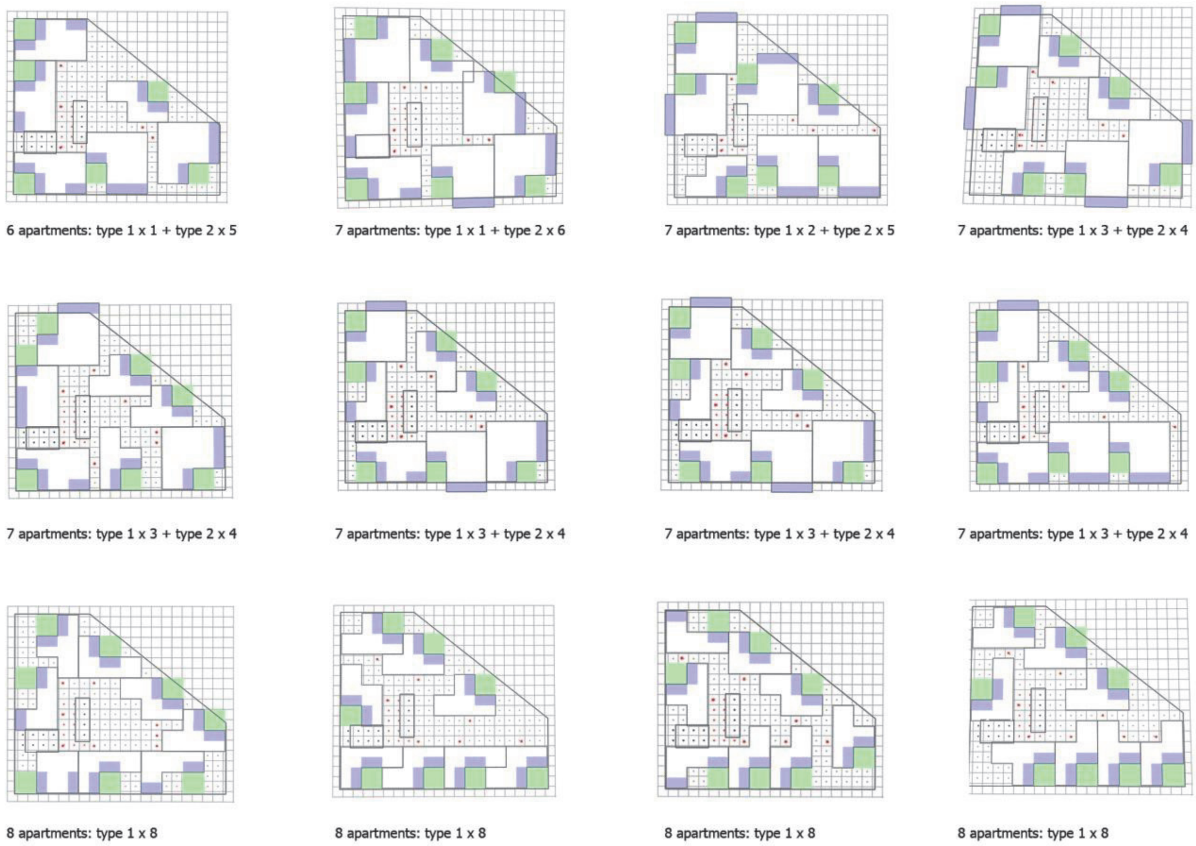


***Fig235**. Dudley House, CDR, 2009: twelve resolved floor layout configurations with different apartment mixes; because of the triangular footprint the design team also built in left-over spaces between apartments and perimeter*

The GA uses the roulette wheel selection choosing from a population of 5000 individuals, with each parent selected separately for breeding. This guarantees a broader gene pool and helps to avoid local maxima and genetic drift. The cross-over procedure cuts the chromosomes at two points for higher probability of transforming apartments that do not settle on the grid (i.e. trapped between apartments or in a corner of the perimeter shape). Both cross-over and mutation allow for apartment types to be swapped in and out of the apartment layout catalogue when not successful. This means that the evolutionary search not only packs the floor with apartments set by the observer but evolves the accommodation schedule itself, providing the necessary mix across all floors. Additionally, the apartment layout catalogue is refined by distilling the exact entrance locations for each apartment type.

The GUI visualization only showed the best performing individual of each generation. The update of floor layouts at each generation, given the fittest individual changed, provided the designer-observer with the necessary visual feedback to comprehend the heuristics used by the algorithm to search for good layout solutions. The designer as observer was however confined to watch the evolution unfold and inspect its visualized performances in the application GUI during runtime. The model took the performance evaluation decisions for the observer who set their objectives but set no explicit quantities to optimize towards, allowing the algorithm to develop the mix and configuration according to system-internal behaviours associated to user-occupation affordances.

The Dudley House model remained a proof-of-concept and was not applied to the project submission.

7.1.3 Field as Performance Monitor

The field in Emergent Circulation consisted of a simple partition grid changing in every configuration. Performances were depended purely on the current configuration state. Accessible Assemblies provided some static elements as agencies of KPIs (access, orientation) to evaluate the performance of the configuration states. The third and main case study of this section uses a double layer of field performances that an object configuration needs to measure its fitness against: a passive analytical map providing static performance constraints and an active generative configuration producing dynamic performance constraints. Both performance constraints represent the KPIs that the observer-designer sets as implicit ratios, not explicit values. Explicit quantities are given via development quantums as input. The model evolves to approximate solution states that fulfil the performance constraints locally not globally, meaning that a local distribution of conditions in the field represent the global solution. The field consisting of static and dynamic performances represents the fitness monitor.

Smart Solutions for Spatial Planning (SSSP) introduced in 6.2 consisted of six simulation models to compile a digital chain for spatial planning. The fifth model – Urban Mix & Density (UMD) - aimed at generating land-use mix and density solutions. The models of 6.2 for primary and secondary street networks generate the partitioning of the regeneration site into development plots. Development plot polygons, existing contextual and generated street network serve as an input site for the development quantum to be distributed across. All street segments and plot cells have been pre-analysed for accessibility using the Dijkstra algorithm to evaluate primary and secondary access points (public transport, shops and other public facilities) and for proximity to site conditions like water edges and motorways. The distances are calculated on the same mesh used to generate the circulation networks of 6.2, which was set by the planners as an orthogonal urban infill grid and triangulated for shortcut diagonals between vertices (square + diagrid). On each edge one land-use cell is allocated sampling desired conditions at that position. A hybrid representation is therefore used that crosses the territorial mesh with generated network graphs (Derix el al. 2012).

***Fig236**. Urban Mix & Density model, SSSP, 2008: (left) the output of the primary and secondary urban network provides the developable plot outlines for the UMD; (right) a settled solution showing four different land-use nodes distributed across the site; clearly visible are emergent clusters of uses and areas of differentiated mix*

The observer input is given in the form of the development quantum loaded via an XML file format including targets for:

- number of units / land-use — m² (target area/grid spacing = units)
    - residential — blue cells
    - commercial — red cells
    - retail — grey cells
    - green space — green cells
    - office high-rise — black cells (only for DavisLangdon study)
- adjacency matrix between unit types — ± metric distance weight
- proximity of units to site conditions — ± metric distance weight
- accessibility to services — ± metric distance weight
- min/max floors per land-use — 0 (parks) – *n* floors
- traversability (i.e. parks) — Boolean (true/false)

Given that the mesh had several thousand edges with a quantum on average of 1000 units, the number of combinations fulfilling the constraint set would be very large and NP complete. It was felt that the standard GA would not be effective for this type of complexity and a different meta-heuristic was found by Åsmund Izaki of CDR, designed to avoid local minima across a solution landscape with many possible minima. The UMD is based on the Quantum Annealing (QA) meta-heuristic (Das and Chakrabarti 2005), itself a derivative of Simulated Annealing (SA - see 3.1.1). Unlike evolutionary algorithms, there is no embryology or evolutionary mechanisms like cross-over but QA follows the principle of evolving populations by a kind of mutation and a cost function (fitness) evaluating its individuals (phenotypes). The heuristic is based on concepts called *tunnelling* and *swapping*: The *n* number of swaps in each individual configuration (solution) is a function of monotonically decreasing time, called *energy* (*temperature* in SA). Swaps are the literal swapping of land-uses on the mesh. Initially this number is large and decreases over time. This heuristic is akin to a dynamically decaying mutation rate. All individuals in a generation apply the swapping and evaluate all local units' compliance with their target constraints, producing a global performance score. The winning state initially selects the best performing individual across the whole solution landscape (generation), which is called *tunnelling* as local minima can be crossed beneath. Over time, the energy of the system decreases and the jumps become smaller as the differences in performance from one generation to another decrease and less swapping takes place within each individual. This heuristic is reminiscent of *hill-climbing* (see 3.1.1) across a solution landscape while avoiding local minima. The notion of 'quantum' refers to the existence of many possible simultaneously minima.

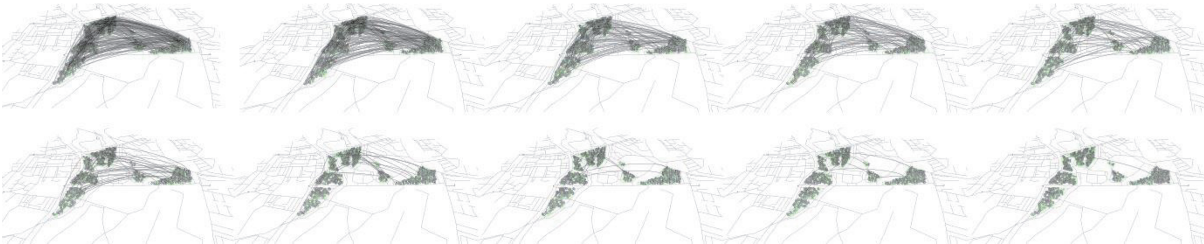


Fig237. *Urban Mix & Density model, SSSP, 2008: ten extracted generations of the swapping process at an individual configuration level, showing how the energy to swap decreases over time as the configuration settles*

The initial condition is simply the random distribution of all land-use units on the mesh edges. The swapping process allocates the land-use units in their favoured condition dynamically. The target compliance evaluation is computed for each unit in every generation, calculating the Dijkstra shortest distance for all weighted connections. Hence, the global performance fitness is the sum of all local field conditions:

$$individual.fitness: \sum_{i=1}^{n} If\,(use(i).cell(j).constraint(k) == true)\; Then\; invidiual.fitness \mathrel{+}= 1;$$

In other words, the fitness score is equivalent to the number of cells for each land-use that comply with the adjacency constraints (of which there can be many simultaneously). Because the configuration changes through continuous adjustment of land-use allocations, the constraining field as well as the configurations co-evolve simultaneously.

The swapping heuristic is a behaviour that corresponds to a designer's behaviour. While the analogue heuristic would evaluate conditions through a different method by probably swapping units serially one by one, the observer-designer can identify with the general algorithmic concept if visualized appropriately. To communicate the behaviour of a QA, it is important to visualize the *energy* of the system states, which can be done by showing the swapped land-use configurations of the best individuals of each generation. It shows that the model learns initially fast and then fine-tunes local areas into what appears to be more detailed mixes on each development plot, aligned with the order of analogue design stages.

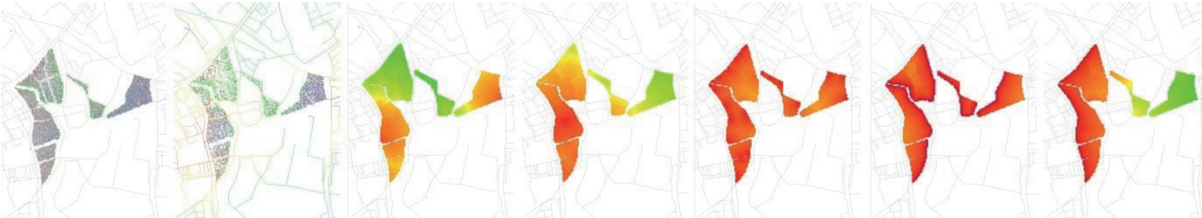


***Fig238**. Urban Mix & Density model, SSSP, 2008: site evaluation criteria of an individual configuration (left), showing (from second left) the access distances to public transport for the road network edges, the walkable access times for each land-use position, access to secondary access points like shops, walkable site access, proximity to public water side and proximity to motorways*

A first prototype of the UMD was shown in section 6.2*,* where many aspects of the urban form were integrated into a single generative simulation model including *betweenness* centrality. As will be discussed later, the integration of many aspects of the built form based on disparate performance criteria as shown in 6.2 was not appreciated by planners and urban design consultants who partnered in SSSP. It was felt that an over-constrained and over-integrated behavioural model could not be visualized intuitively. The narrative that stakeholders identify their heuristics through would be lost and consequently they could not approve resulting configurations. Hence, the second prototype of UMD discussed here uses a leaner more parsimonious behavioural representation based on performances corresponding to design heuristics (Derix et al. 2012).

However, because underlying quantities were encoded anyway, a layer for building heights was added to allow a quick visualization of density levels. Building heights were determined by the min-max range of input constraints and in addition each land-use unit queried its immediate neighbourhood for density by averaging the floor numbers of all eight adjacent neighbouring land-use units. Depending on the neighbourhood average floor height, each unit adjusted its height up or down within the permissible range, reminiscent of a cellular automaton state transition function. The height at each grid position produced a good indication for the eventual mix and density level for the combined development plot. This was supported by the walkable access criteria for each land-use unit and street network edges comprising a plot polygon. All field conditions providing static and dynamic constraints could be toggled and visualized individually during runtime, allowing the observer to further understand the decisions made by the model. Without just emulating the representations and behaviours of analogue design heuristics, the UMD model realized a high degree of identification by the stakeholders. The semantics of planning guidance terminology could be applied to the visible dynamics and results, such 'urban grain, 'mix & density', 'plot ratios', 'proximities', 'walkability', etc (see Aspects of Built Form in ByDesign (CABE 2000)). The consortium came to the conclusion that the UMD has the potential to realign planning phases of the general urban regeneration workflow because traditional zoning diagrams could be made redundant. At an earlier stage of the workflow UMD can generate a relatively detailed mix and density levels in tandem with the weighting of the development quantum. This is commonly done in two or more stages, divorcing the above mentioned dependencies into separately resolved aspects.

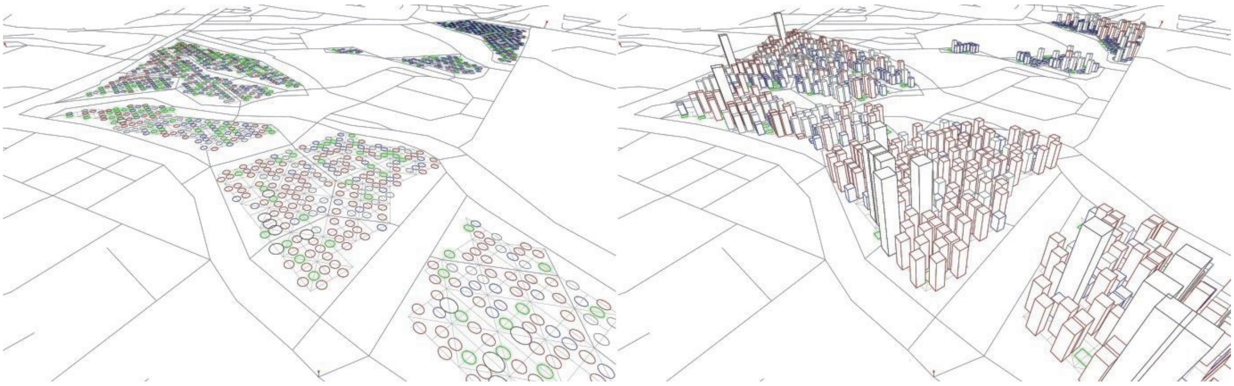


***Fig239**. Urban Mix & Density model, SSSP, 2008: (left) the hybrid grid and road network input of developable plots with a distributed mix of land-uses; (right) the height and density visualization of the mix showing diagrammatic clusters of land-uses in different areas of the site subject to site constraints, including an office high-rise typology in a case study for the developer DavisLangdon*

7.2 SITUATED OBSERVER | EMBEDDED AGENCIES

The remote observer instrumentalized field performances for explicit evaluation, separating meta-heuristics from analysis. Performance evaluation occurs in isolation from the generative process with the observer-designer setting targets for performance fitness at the start of the configuration process. Although performance fitness is calculated for each individual in a generation and even local position within an individual, the overall fitness determines the 'goodness' for selection, meaning that well resolved sub-areas of a configuration are ignored if the individual is not selected. In order to take more control of a configuration during generation, the configuring process requires more openness without reverting back to a completely analogous assembly. At least two approaches serve to provide more control of the process:

a) the observer-designer is situated within the developmental process rather than remotely watching the configuration unfold
b) the analytical evaluation model merges with the generative heuristic model

Situating the observer-designer means that the generative process must be less automated across generations. Evolutionary algorithms can be made interactive through artificial selection but the runtime processing is mostly computationally very heavy and event handling (the event of a user selecting some geometric element and changing its state) very complicated to integrate (updating changed elements).

EAs are also quintessential automatic optimization models and are not ideally suited for participation, unless the observer engages in the selection process as shown in section 5.1, which allows for direction but not participation. In other words, non-teleological processes are required that are not aimed to steer towards some optimum state but towards a consensus with the observer-designer. Bottom-up processes were generally considered by Paul Coates and John Frazer to be consensual (Coates 2010) but only as a remote-observer model. In a professional design setting this observer distance does not make sense for conceptual design. Bottom-up automation works well when concept and quantities have been decided.

When a design search takes place within the pre-concept stages without a geometric formalism, the model should not be generation-based and must not be population based. No sequence of events is determined as in a parametric model which specifies both parametric dependencies as well as event sequence, i.e. the iteration of construction. The computational model must be as open as possible while still performing vital generative and analytical tasks. The observer-designer must become a third component, negotiating areas of the design search space that were before controlled by the meta-heuristic. Previously, the observer's intentions were explicit in fitness functions but now intentions need not be explicit but aligned with the heuristic of the generative process.

Being solely a heuristic non-teleological generative process, the configuring model must include some constraining on itself to provide intentionality. The models presented here try to erase the separation into generative and analytical algorithms by allowing the analytical process to drive the generative heuristic, even turn into the configuring process itself. Targets are replaced by behavioural performances of the field. Those behaviours that usually evaluate states are made explicit so as to become operational diagrams. As a consequence, behaviours of the observer-designer interacting with the model without explicit targets as well as the behaviour of the performance-based generative heuristic need to adapt to each other in runtime. Compliance testing evolves to be behavioural consensus.

The observer provides two agencies now: the agency of the designer and the agency of the user-occupant. As a designer the observer selects the generative model driven by performance states. Because there is no target state, the designer needs to take the role of the observer-occupant whose agencies are also represented through interaction with the model to adjust unresolved areas from the occupant's perspective. The designer must always act *as if* he was also the user in the represented spatial environment. In the Remote Observer, the observer acted only as designer. Now, any configurational state of the model must reflect the behavioural consensus between heuristic generation and dual observer.

The two projects discussed in this section use the abstractions of chapter six (map, graph, network) to drive the generative heuristic process as operational diagrams.

7.2.1 Heuristically-Driven Configuration

The Future of Construction project collaboration with Fraunhofer Institute was introduced under Space-Behaviour Correlation (FuCon – 6.2). The model discussed was the third of three algorithmic planning simulations for programme allocation on floors according to circulation (now called FuCon3). The edge-bundling algorithm provided a clear correlation between designing configurations and heuristic behaviour of the algorithm and thus a good example of knowledge generation based on an analytical representation. FuCon3 analysed for betweenness centrality as a performance criterion of the configuration and proposed a morphological state derived from the nodal set placed by the observer. The second algorithmic planning simulation also proposed an edge-bundling model for the second planning stage - the building programme distribution and massing (FuCon2) – but aimed to be less dominated by the algorithmic heuristic to allow for more participation by the observer as agent of design and occupation. FuCon2 uses the second-order abstraction of graphs, operationalizing the representational diagram to let the observer inform the state of the configuration. As a consequence, the graph representation re-organizes the local aggregates into accessible functional areas. As opposed to FuCon3, the operational graph can be differentiated locally to enable refinement of areas within the building.

As such this case study is still driven by the heuristic algorithm yet no evaluation for optimal states is conducted. The observer needs to negotiate performance states with the model *as if* situated within the building, not distinguishing between generation and analysis. Performance states are generated from consensus between algorithmic model and dual observer.

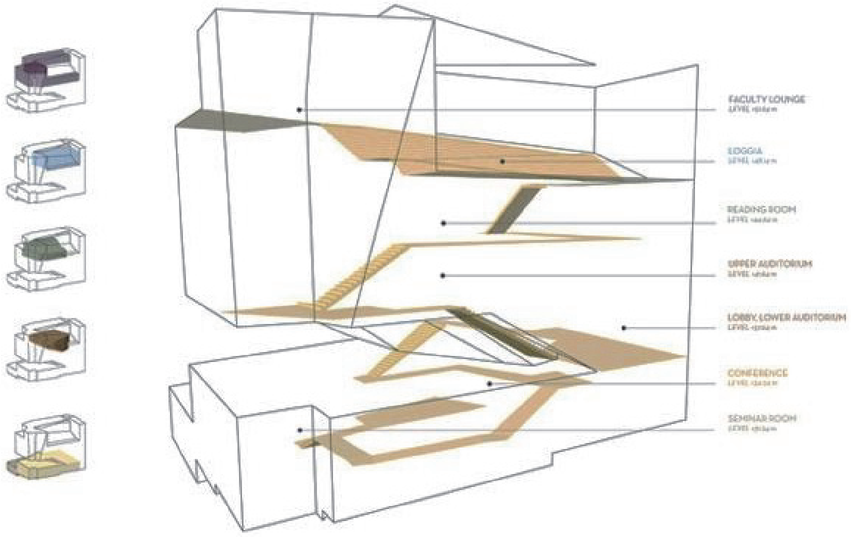


***Fig240**. Issam Fares Institute, office dA, 2006: the design concept adopted for FuCon 2 is derived from the analogue design heuristic of 'carving' the circulation out of the building volume and grouping the functional room clusters around access points*

FuCon2 was developed by Lucy Helme to generate building massing by distributing the accommodation schedule within a given envelope. It was decided that the design methodology to distribute the building programme should like FuCon3 be driven by the circulation structure across the building volume. The circulation was thus meant to carve out semi-public movement spaces that provide a visually legible circulation. The algorithm to translate this methodology was also identified to be an edge-bundling algorithm albeit a variation of FuCon3. This choice was no coincidence as the whole of the FuCon algorithmic demonstrator aimed to show that computational heuristics can be designed with an equally coherent design concept across multiple scales like a traditional building design. The approach was called *algorithmic consistency* to reflect self-similarity across scales not as a morphological but a methodological concept (Krause et al 2011, p452). Further, circulation in spatial environments is meant to provide a cognitive scaffold for way-finding and in the case of workplace buildings facilitate informal communication[63] (the case study

[63] "*Staircases to be located for building users to have the option to use them over lifts, with the design of the staircases providing visual connection and social interaction opportunities*." (BCO 2009, p44)

building was a hybrid building of laboratory and offices). Hence, edge-bundling was perceived to be a good choice of algorithm as it organizes a topological connectivity graph into efficient flow structures and simultaneously reveals locations of encounter for informal communication.

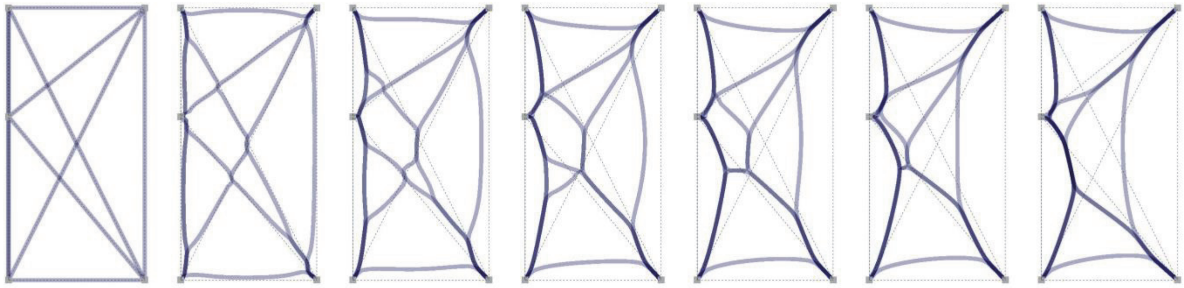


Fig241. *FuCon2, 2010: stepwise visualization of the edge-bundling process in seven frames as a demo in two dimensions*

The initial graph represented a simple connectivity graph in three dimensions with nodes being the access points to building programme (room cluster centres) and edges connecting those nodes based on an adjacencies matrix. Both accommodation schedule with room clusters and adjacency matrix could either be imported via a comma-separated-value format (csv) or manually compiled in the GUI. The force-directed edge-bundling used here does not require any pre-processing like the hierarchical mesh-controlled edge-bundling of FuCon3, which calculated a Delaunay triangulated mesh as a control structure (Holten and van Wijk 2009). It is a hierarchy-free representation and self-organizes the bundling process through spring-forces (Helme et al. 2014). All connection edges of the topological graph are subdivided into equal segments and vertices interpolated. Edge pairs are checked for compatibility for bundling based on restrictions on the angle between edges, difference in length and remoteness. Edges that fulfil such pairwise restrictions find the equivalent interpolated vertex and a simulated electrostatic attraction force pulls them towards each other. As described in section 5.2, the force used to attract vertices is proportional to distance according to *Hooke's law*. When equivalent subdivision vertices attract each other, the edge itself needs to adapt in length and direction of the attracting segments. This is done via the spring analogy: all consecutive vertices along an edge are represented via a spring, that is, each segment is controlled by an attract-repel force that is dependent on the local stiffness value, itself calculated from a global spring constant.

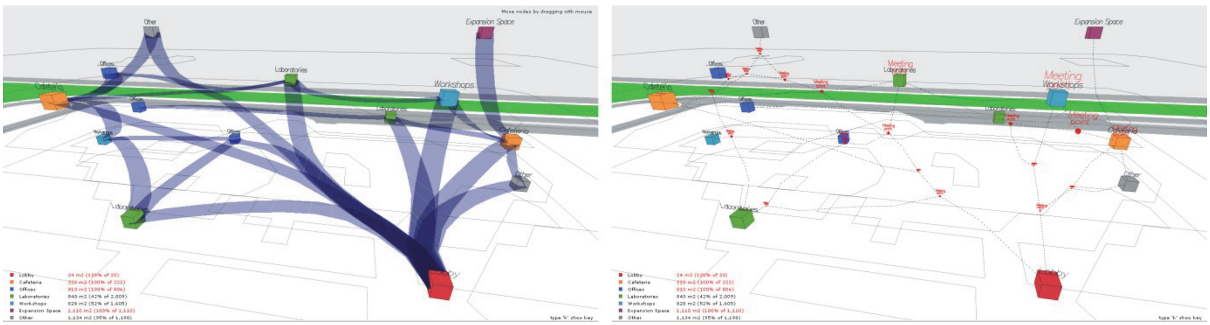


Fig242. *FuCon2, 2010: (left) the edge-bundling of a connectivity graph of the adjacency matrix rendered as blue ribbons; and (right) the new connectivity graph resulting from the interpretation of the edge-bundling, showing the proposed new programme areas as red dots;*

The visualization of the topological graph into the bundled circulation is achieved stepwise, so that the observer can follow the attraction across edges contracting the graph into flow bundles. The observer can chose to interfere at any time with the process by freezing the bundling to interaction at the node distribution level, recalibrating the circulation diagram. Unlike any known application of edge-bundling, the circulation diagram is generated in three dimensions producing limited locations of intersections of more than two edges. Those locations are highlighted for potential additional programme such as informal communication spaces. A new hybrid connectivity or circulation graph results from room cluster access points and encounter areas. Interpolated areas of encounter result from the simplification of the edge-bundling into a straight edge graph from which graph cycles can also emerge that represent circulation loops. The observer can toggle between the active edge-bundling visualization and interpreted circulation graph with additional communication spaces (Fig242).

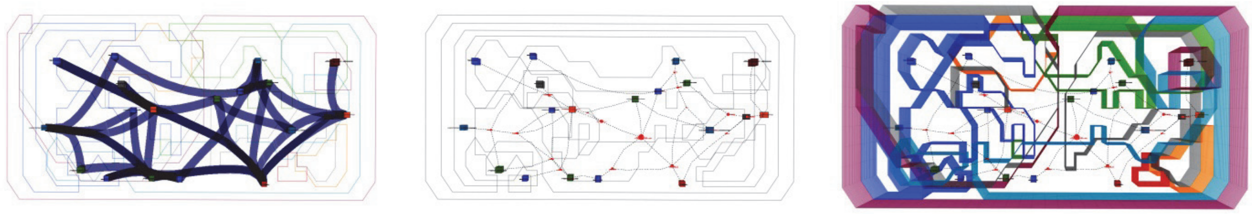


Fig243. *FuCon2, 2010: top view of (left) the edge-bundled adjacency graph; (middle) the interpreted circulation with interpolated encounter areas for new programme and (right) the partitioning for functional areas diffused around room cluster access points*

The building programme of functional areas in m² is diffused radially from the room cluster access nodes onto the floor to which the node is specified. A simplified reaction-diffusion algorithm is used to partition the programme areas. Functions are rendered as polylines in colour. The diffusion might exhaust itself before reaching the envelope or a neighbouring programme and thus can generate porous floors representing redundancy in the accommodation schedule, called space-left-over-after-planning (SLOAP). Additional to the programme area diffusion, the circulation diagram is carved out of the envelope-inscribed volume to produce the final building massing. This is done via an invisible grid whose nodes are tagged by proximity to the circulation graph and deactivated to void the volume around the circulation, in accordance with regulatory dimensions for circulation. The rendering methods for the visualization were not refined, which was left to the VR system of the Fraunhofer Institute.

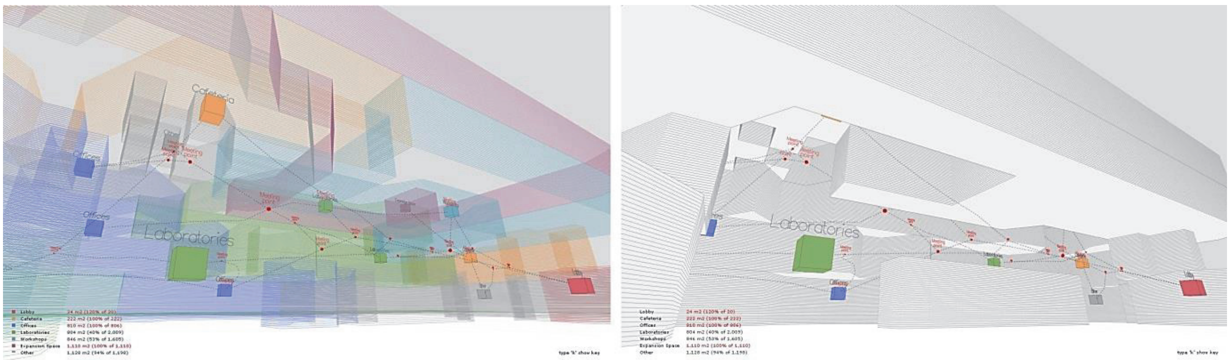


Fig244. *FuCon2, 2010: (left) the area programme diffused around the room cluster access points rendered as coloured polylines for transparency; and (right) the massing resulting from 'carving out' the emergent circulation diagram from the programme area and envelop volume*

The dual observer as designer or potential occupant can manipulate all aspects of the configuration during runtime. While the topological graph generating the circulation diagram via the edge-bundling heuristic is constantly processing, interactive interference is absorbed instantly by the algorithm and the impact rendered visible without optimizing the configuration. The spring system in fact, does not immediately resolve into the most efficient flow bundles but due to attract-repel forces produces some delay in settling, giving the observer time to understand the projected circulation diagram. Thus, the algorithmic and observer heuristics are associated by concept and behaviour allowing full participation by the designer in the search process. Unlike generation-based optimization, the edge-bundled diagram can be manipulated locally. When a room cluster access node is moved, only the flows affected by this node are updated, leaving resolved flows intact.

Fig245. *FuCon2, 2010: the FuCon demonstrator set up at the BAU 2011 building expo in Munich, 2011; the author designing a configuration in a tracked and immersive 3D VR environment developed by Fraunhofer Institute*

Although the force-directed edge-bundling itself does not constitute a spatial evaluation, the result of the algorithmic heuristic is known to produce efficient flow graphs akin to minimal spanning trees and minimal path networks proposed via an analogue model by Frei Otto (Otto and Rasch 1995; Otto 2005). Hence, while path analysis algorithms such as Dijkstra's shortest paths or network analysis measures have been used previously to evaluate the performance of the generative process, here the evaluation and the generation are identical. The observer must instil the purpose of the design concept through interaction rather than constraining the model via targets. To guide the observer-designer with performance states, functional area types are read-out in the GUI for compliance.

7.2.2 Empirically-driven Massing

Based on a live project at Aedas London, a heuristically driven model was developed that extended the two component models of FuCon (analysis correlating generation) into a three component model (analysis correlating generation and evaluation). For the Euston Crossing feasibility project in 2013, CDR was primarily commissioned to support spatial analysis. Euston station and its surrounding are undergoing several planning exercises to host the new High Speed railway between northern England and London (HS2). TfL commissioned Aedas to review the design of Euston station forecourt, including Euston Square and the pedestrian crossings of Euston road. The feasibility study was meant to produce a report into for the rearrangement of the square, bus terminal and potential new commercial massing on the forecourt. CDR conducted an accessibility analysis based on data collected by previous consultants like Space Syntax Ltd, Arups or Intelligent Space and produced 3D visibility analysis for the exposure of proposed massing options.

Fig246. *Euston Station, 2013: current spatial condition in front of the station, prohibiting clear orientation for visitors and commuters (from Aedas report for the TfL Feasibility Study)*

The approach by the design team of separating access analysis, visibility exposure, context scale analysis and massing design felt archaic and a pilot for an alternative approach via computational simulation was proposed. Like FuCon2, the Euston Pilot was not meant to be an automated optimization model as the quantities for optimization would result in non-compliant or non-sensitive options for massing. The commuter to and from Euston station was meant to experience a seamless transition from the station forecourt into his local context and the massing itself was meant to have little visual impact on the global context, including some elements of the London View Management Framework[64]. In other words, a model was sought that drives the massing from empirical associations between a building performance through GFA and envelope, visual impact and contextual experience of scale and access. The algorithmic heuristic would therefore have to emulate the observer-occupant *as if* situated in the city through integration of the observer-design heuristics.

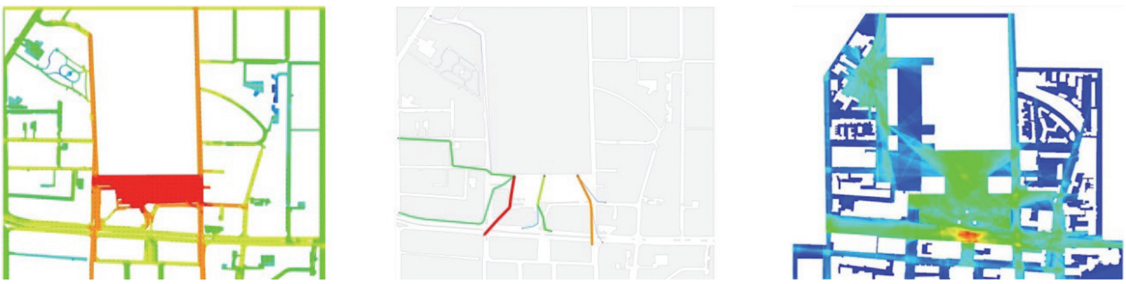


Fig247*. Euston Crossing, 2013: three modes of analysis conducted by CDR for the Aedas London design team; from left to right, access levels and flows (here topological access), movement routes and footfall and visual choice*

[64] London planning prescribes certain view axes and panoramas not to be impacted by new developments: http://www.london.gov.uk/sites/default/files/archives/LVMF%20low%20res%20part%201.pdf, accessed 10.11.2014

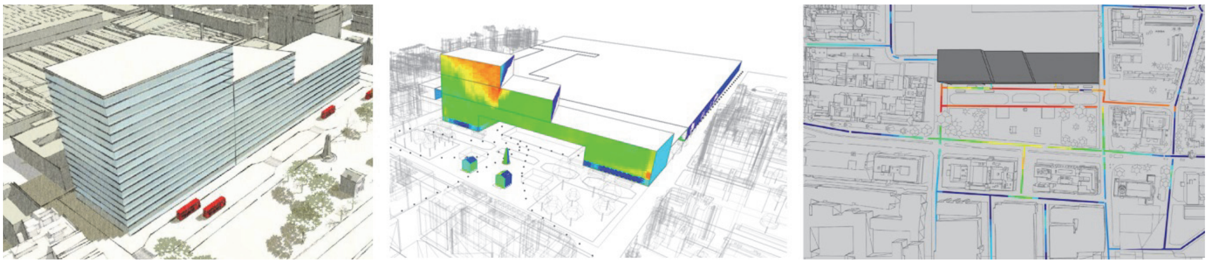


***Fig248**. Euston Crossing, 2013: massing design options analysis conducted by CDR for design team, showing (left) a massing option, (middle) visual exposure of the envelope and (right) visibility from access routes*

A three component model was set up to approximate an empirical dual observer design simulation, comprising

a) pre-processed analysis;
b) run-time generation based on analytical and interactive drivers and
c) post-generative performance evaluation.

The first component calculates the visual choice values as a discretized map of the site extents, using the Visible Polygon Traversal Algorithm (VPTA – 6.1). Visual choice is based on what was called the *openness* measure in the Objectives-to-Measures-to-Perception Correlation table, constituting a ratio between visible and hidden edges of an isovist. The concept of *choice* provides an indication of how many adjacent spaces are perceived at a location, which in turn provides choices for movement directions. Isovist openness values are encoded in plan and loaded as input layer into the application, encoding number of open edges visible by location in a CSV file format. In relation to railway stations, visual choice for movement is a well-known orientation issue for commuters and visitors who exit a station and search for directions towards their destinations. Particularly at Euston station, this is regarded as a major problem where way-finding in front of the station is impeded by many disjoint elements.

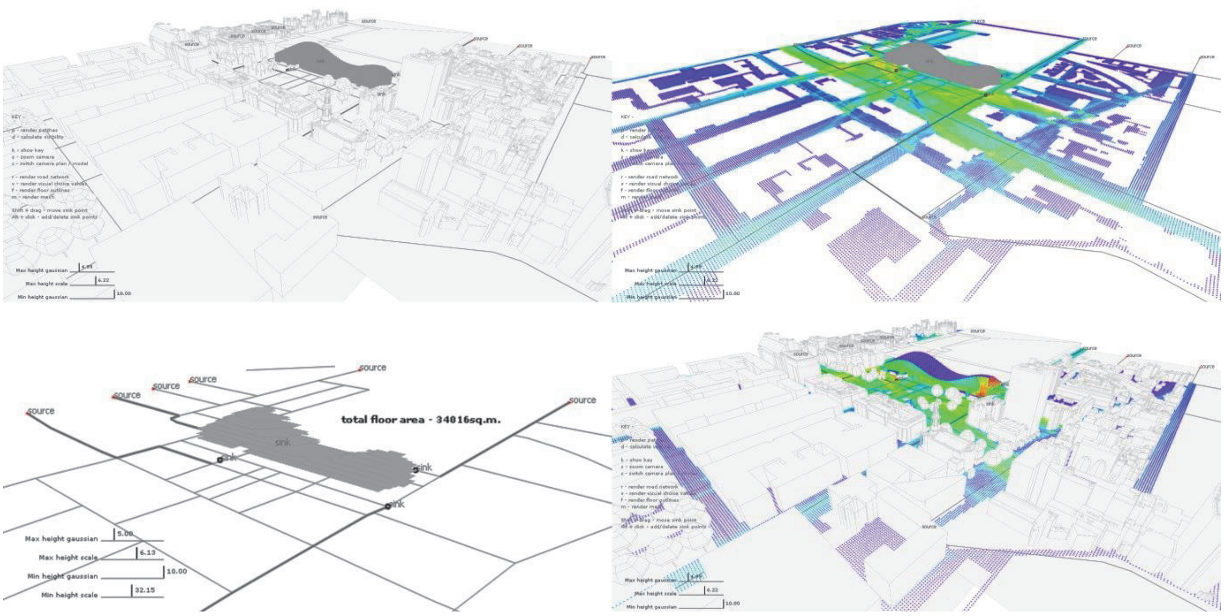


***Fig249**. Euston Pilot, 2013: (top left) the site model within GUI; the grey massing represents the scaled patches of the proposed building footprint; (top right) the first component: visual choice map; (bottom left) the access points and their shortest route graphs; also visible are the four Gaussian distribution sliders and the floor representation of the building mass and its current GFA performance; (bottom right) the visual exposure analysis for the current massing state*

The second component comprises two types of interactions: the manual positioning of access positions to the massing and the weighting of the effect of access positions onto the massing. Both interactions trigger algorithms that evaluate quasi-empirical quantities and translate those into the proposed building mass. The interactive positioning of access positions as destination points to the proposed building, recalculate all OD pairs via the Dijkstra algorithm. The origin points are placed in the input drawing and have been inserted at all street ends of the site perimeter for the pilot, assuming that commuters can go/arrive from anywhere. The model calculates the Dijkstra graph from all origin to nearest destination points and thus creates one or more tree graphs. When the observer interactively moves the access positions, all OD pairs are simultaneously updated. Apart from all shortest routes to/from the proposed building entrances, the algorithm samples the street sections along the edges of the graphs. This is done by averaging the building heights adjacent to the graph edges. Each access point receives a value for the average context scale experienced along the routes compiled by the graph it connects to. This context scale value is projected onto the nearest perimeter point of the proposed building, i.e. the access point does not have to be on the perimeter but only near it.

Similarly, the visual choice value at the underlying grid position of the access point is projected onto the nearest perimeter point of the proposed building. Thus, the access point stores two empirical parameters about the context: contextual scale and orientation. To control the impact of the contextual parameters four sliders are available in the GUI of which two help to guide the impact of each contextual parameter on the massing. The two sliders per parameter control a Gaussian distribution and the scaling of the distribution. The proposed building footprint is subdivided into discrete patches and for each access point the nearest patch is established as 'winner' inheriting the context parameter values. The Gaussian distribution and its scaling determine the height of the building mass at each patch within the footprint perimeter. When the Gaussian distribution increases, more neighbouring patches of the winner patch are likely to share the context parameter values and thus increase in height. Inversely, the smaller the distribution value the tighter the sharing of the context parameters and patches outside this distribution tend towards zero height.

For the visual choice parameter, the distribution controls the size of the entrance patches by adjusting the lower z-axis value: if the visual choice value is high at the access point, the opening of the entrance size at the building perimeter is decreased by the distribution since the site in front of the entrance facilitates orientation. And vice versa, if the visual choice value is low at the access point, the entrance size is increased to allow commuters to see more context for orientation while exiting. Dijkstra shortest routes and Gaussian distributions are activated by the observer to simulate likely movement and visibility performances as effects of the environment onto the building mass. The algorithms are representative of 'good practice' design heuristics and map otherwise empirical contextual quantities directly into massing options as design drivers. The border between analysis and generation is erased.

Finally, the third component represents an explicit performance monitor by allowing the observer to toggle visualization of the proposed building mass from envelope done through scaled patches to floor outlines. A simple gross floor area calculation of the floor outlines provides some feedback on the area performance associated to the configuration of the model (access point positions, route scale experience, visual choice and distributions). The observer can also evaluate the visual impact of the negotiated mass onto the road network from which the Dijkstra algorithm calculates the sub-set of shortest routes. The patches are rendered in the colour of the exposure value described in chapter 6.1. The combination between numerical target evaluation and visual impact evaluation guides the decisions for interactions to be taken by the observer-designer, closing the process loop.

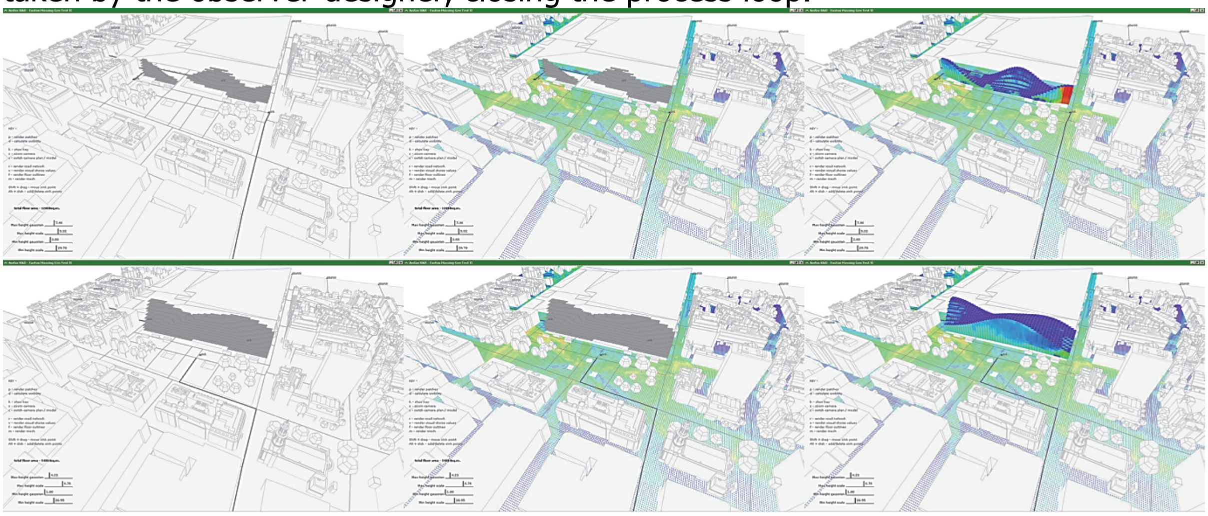


***Fig250**. Euston Pilot, 2013: two massing states by changed access points and Gaussian distribution weighting producing very different building mass performances*

All three components use spatial analysis to represent correlating aspects of human perception and behaviour in the environment. Visual choice and exposure use discrete position values as first order abstractions, activated by the transformation of behavioural diagrams of the movement graph as second order abstraction. Interaction and Gaussian distribution negotiate the associations between perceptual, behavioural and spatial performances. The observer as designer has a clear agency that does not simply put targets for optimization or interpreting analysis after the fact but negotiates design heuristics with occupant empirical heuristics (for example way-finding behaviours). Building mass and area emerge from the associative mapping of the spatial environments' performances onto the building plot, which the observer-designer mediates by interacting with the algorithmic heuristic. Mediating analytical and heuristic algorithms removes the necessity to build artificial schemata containing hierarchies of parametric dependencies. Associations are less deterministically formed and the design workflow is more open than an iterative process as proposed by KBD and parametric modelling. The lack of a strict ontology in the Euston Pilot means it is not a spatial configuration that is being produced but the live definition of a minimal spatial element visualized via the associative structure. As Hillier and Leaman (1974) proposed, there are no spatial elements but only commutative structures that give rise to morphologies.

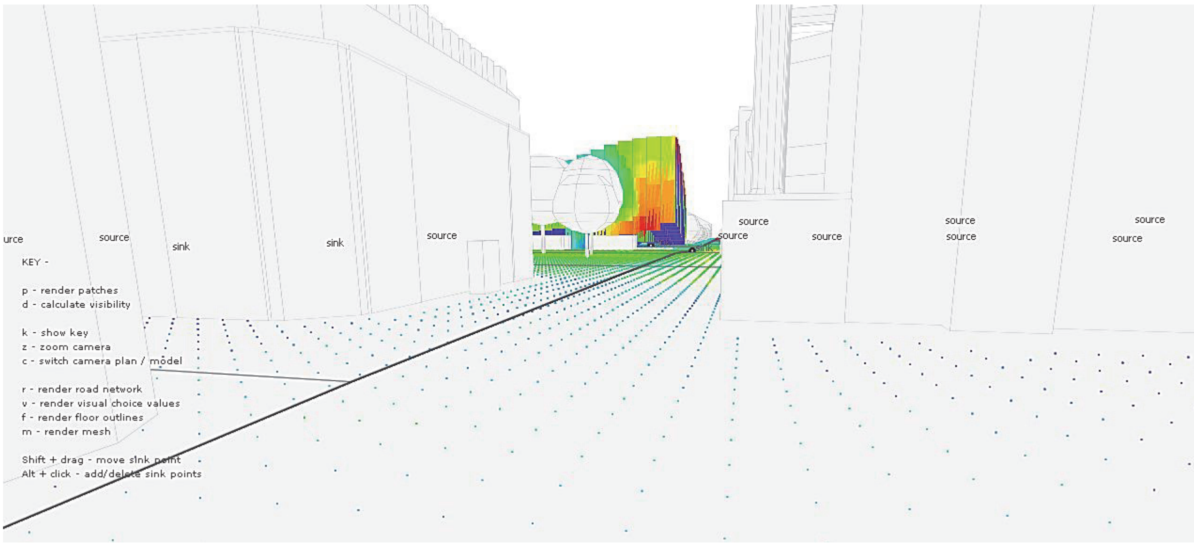


***Fig251**. Euston Pilot, 2013: a view from within the site model looking up Woburn Place towards Euston station with a new massing state on Euston station forecourt*

The Euston Pilot merely represents a test to create a proof-of-concept that the occupant's empirical knowledge of the field can be activated live in a simulation model to generate spatial configurations. The proof-of-concept does however realize aspirations of design theorists such Raoul Bunshoten of CHORA who elaborated site-analysis into a generative approach that James Corner called *Game Boarding* (Corner 1999, p239). The issue with the approach – as could be observed in many architecture schools – was that after a long mapping period, the rules of gaming for the site board were difficult to extract and often arbitrary or *ad hoc*. The Euston Pilot instead follows Kevin Lynch's approach (1960) of working directly with cognitive heuristics of occupants and spatial properties to define urban structure.

7.3 LEARNING OBSERVER | ASSOCIATIVE PLANNING

A generative model based on associations can be envisaged that is not in use either in academia or industry. Based on the third-order abstraction discussed in the introduction to chapter 6.3, self-organizing neural networks by definition remove the iterative layers of weighting that supervised networks require to produce a direct mapping between input and output layers. The single layer map represents a non-hierarchical analytical field of differences and similarities between input features. The SOMs and their adaptive derivatives discussed in chapter 6.3 used the output layer principally as an analytical map to support design strategies and inform briefs. In this section, the output layer is used as a generative map from which diagrams of spatial configuration can be extracted. The map still functions as a difference distribution mechanism but drives morphological parameters. In other words, morphological representations must be related to associative fields in the output layer. Projects will illustrate degrees of abstraction for this relation, attempting to correlate morphological representations to cognitive associations.

If we accept the premise that unsupervised self-organizing neural networks can generate their own ontological schemata by learning empirical dependencies between features, then it can be said that the models presented here use emerging schemata to represent some spatial type. Depending on the class of data that is used as feature input space, the learned schemata represent patterns of buildings as much as patterns of occupation. To reiterate Coyne's insight already cited: "*There is no explicit representation of a schema. However, a schema is implicit in the pattern of associations generated by the system during the learning process*" (Coyne and Newton 1990, p40). Ideally, one might expect that known building typologies usually encoded via accommodation schedules and other standardized representations could be overcome by the use of non-geometric dimensional data describing the use of buildings. Or one could hope that the network might generate new schematic representations from the standard sector data by finding new associations.

***Fig252**. Mood or Inspiration boards for four concepts*[65]

If the associative network is working as a generative mechanism then one might also ask what type of design heuristic it is correlating to. The weighing of differences from input samples and organizing them into categories is akin to the pre-design stage of *mood boarding* (mainly used in interior design). To generate a design concept, the design brief is interrogated for empirical associations when designers are meant to intuitively compose feature categories that describe the brief visually and semantically through empirical phenomena (Gero 1990) [66]. Phenomenal categories provide the input to the design process where dependencies and dimensions between the identified features and their categories are sought. In other words, first the design schema (concept) is worried then an ontology is solved (to paraphrase Stanford Anderson's (2005) analogy of the design process as 'problem-worrying and problem-solving'). The concept represents the embryology to generate phenotypes of a known building genotype or schema and is not related yet to a design context. The concept provides the design rules to create instances of the generated schema. The design process itself represents a heuristic to decode the instances from the schema within the context of the brief and site. Rachel Cruise (2005) compared those two aspects to *tactical* decision making and *strategic* process. In her Dry Stone wall-building research project, she identifies general decisions that are being taken by a builder from experience and the situational

---

[65] http://www.rit.edu/fa/globalvillage/sites/rit.edu.fa.globalvillage/files/inspirationboardsall.jpg, accessed 02.10.2014

[66] John Gero supports a design structure where a brief leads to input quantification that gives rise to a 'prototype' or ontology. From this prototype, concepts can be generated associating information: "*In this way, design prototypes provide a means by which given a little situational information, potentially appropriate concepts are retrieved, and the designer has available a fleshed-out set of concepts that can lead in many directions.*" (Gero 1990, p33)

strategies specific to site, in runtime as it were. Mood boards are equivalent to tactical decisions that interpret the brief in the context of a specific site and the design process to the strategic decoding mechanism of the schema. Associative networks generate tactical mood boards from an empirical input feature space that the design process has to decode into morphological instances. Hence, the heuristics that are encoded through associative networks represent the search for phenomenal expressions of a space and their empirical categories.

Three models of association-driven generative design for space planning are discussed non-chronologically in this section. They are distinct in their level of feature space abstraction, starting with a clear professional schema of a building typology, to a use-based configuration and advancing to schemata search of cognitive organizations of space.

7.3.1 Associative Partitions

In 2006 the author was commissioned by Zaha Hadid architects to develop a design concept for a competition through computation. The competition was for the new headquarter of the champagne maker Piper Heidsieck. Hadid provided the accommodation schedule, adjacency matrix, organizational constraints (specific constraints on staff demands) and a site plan.

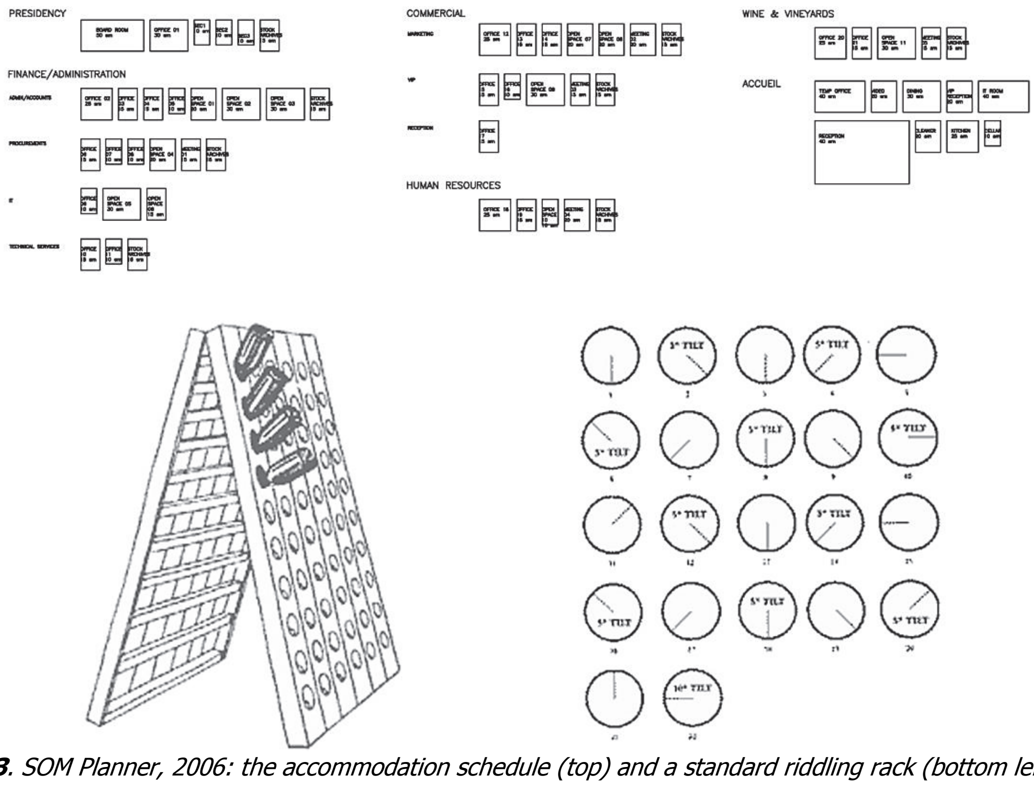


Fig253. *SOM Planner, 2006: the accommodation schedule (top) and a standard riddling rack (bottom left) and the riddling procedural diagram (bottom right)*

The competition provided an opportunity to trial a SOM as a generative planner (SOM Planner). The SOM was identified as an appropriate algorithmic heuristic for the selected concept of the *methode champenoise*. The method is based on in-bottle fermentation for which an elaborate bottle-turning routine was developed, called *riddling*. Riddling can be abstracted into a geometric rules diagram, based on a grid where each grid position represents the direction of a bottle in three dimensions.

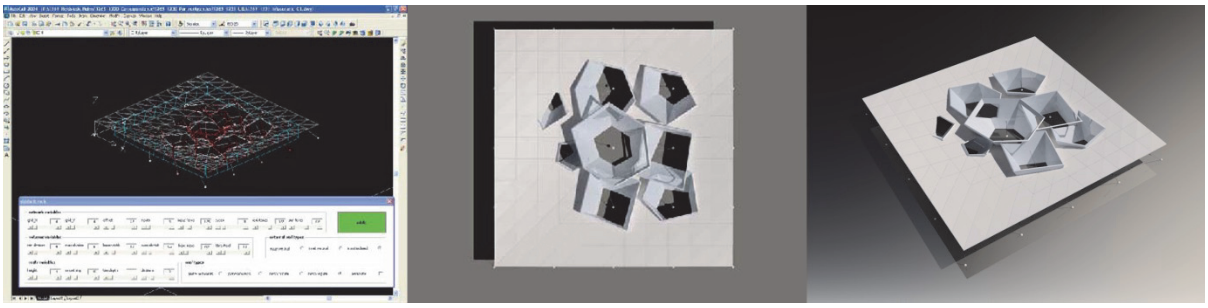


***Fig254**. SOM Planner, 2006: the GUI inside the AutoCAD environment where it was programmed in Visual Basic (left); and the renderings of the solution*

The time-based riddling procedure constituted an analogy to spatial vectors on a grid. The SOM was chosen to organize the directions of the vectors on the grid according to associations to the constraint set from Hadid's input that were encoded into normalized feature vectors. Input vectors representing the accommodation schedule could now be used to train the map using the dot-product comparison method.

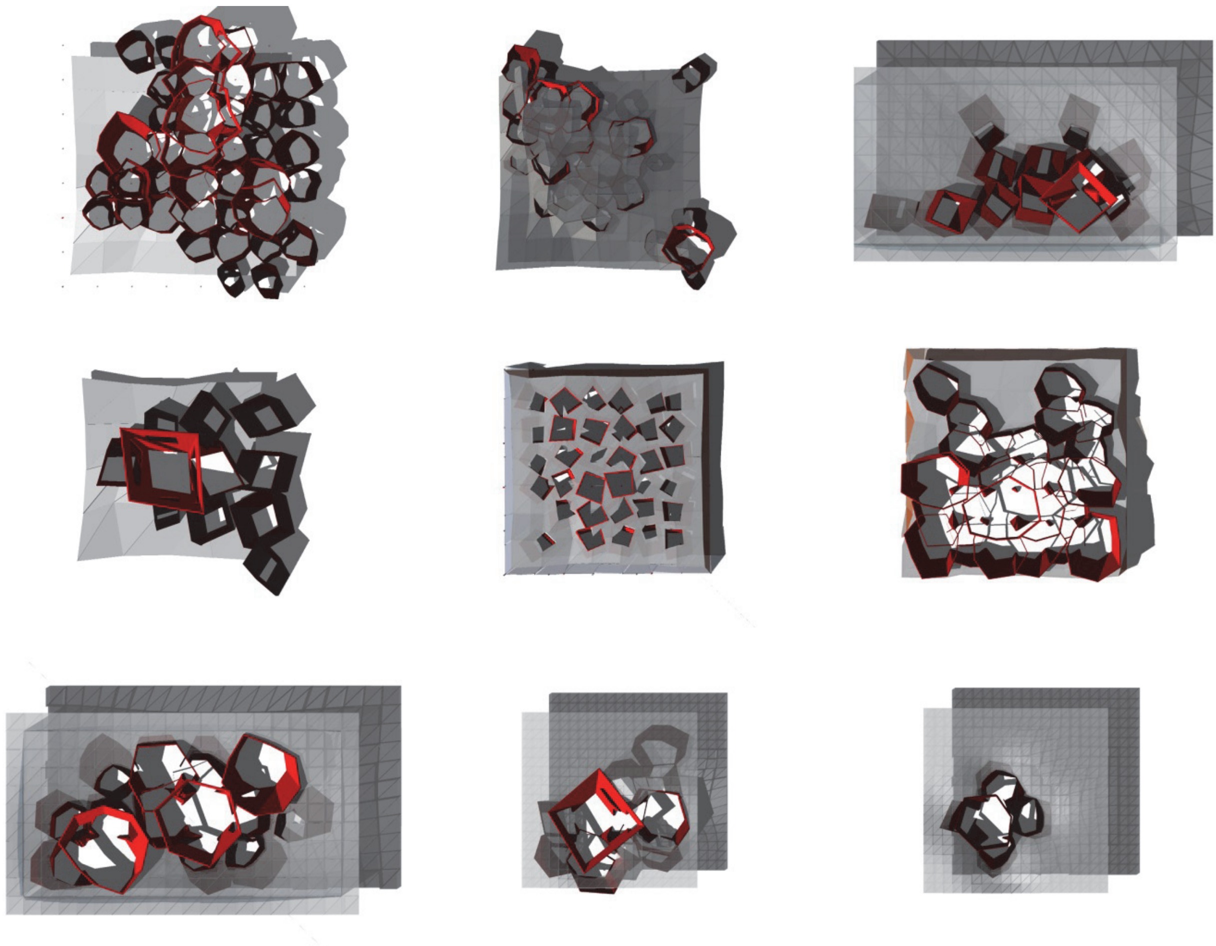


***Fig255**. SOM Planner, 2006: nine generic solutions generated by the author showing how the vectors, their geometrical translation threshold and specifications of the geometric definitions in the GUI produce distinct results*

The dot-product map attributes each room to a map node across a fixed map size, dimensioned to the desired footprint. The geometry of each partition, equivalent to a map node, would be driven by the information of the vector direction and magnitude whose parameters were based on solar exposure values and organizational hierarchy (i.e. the importance of an asset like the CEO's office requires more space, adjacent secretarial spaces and sunlight). Room partitions with node vector magnitude larger than a set threshold would be translated into a cone-like geometry, and those below would become part of an open-space office, as instructed by Hadid architects. All geometric translation criteria could be set in the GUI by the user who weighted features and geometric embodiment.

The SOM as generative planner in this simple example worked robustly with the exception that SOMs on a generic convex grid (see concave shaped maps – 6.3) allocate node winners differently (which maintaining their topology) each time the map is trained and thus a building diagram is hard to evolve as an iterative process. While Hadid architects used the SOM planner to generate initial conceptual sketches, the eventual design was heavily reworked (Fig256). Thus, the SOM planner produces tactical conceptual organizations that the designer strategically translates. The schema of the traditional workplace sector is not questioned by the associative network. Minimal spatial elements of room types are mapped directly as network nodes, maintaining a first-order abstraction of discrete partitions in both the grid as subdivided position and the accommodation schedule as a simple table cell. Associations emerge based on differences on a formal level between geometric quantities.

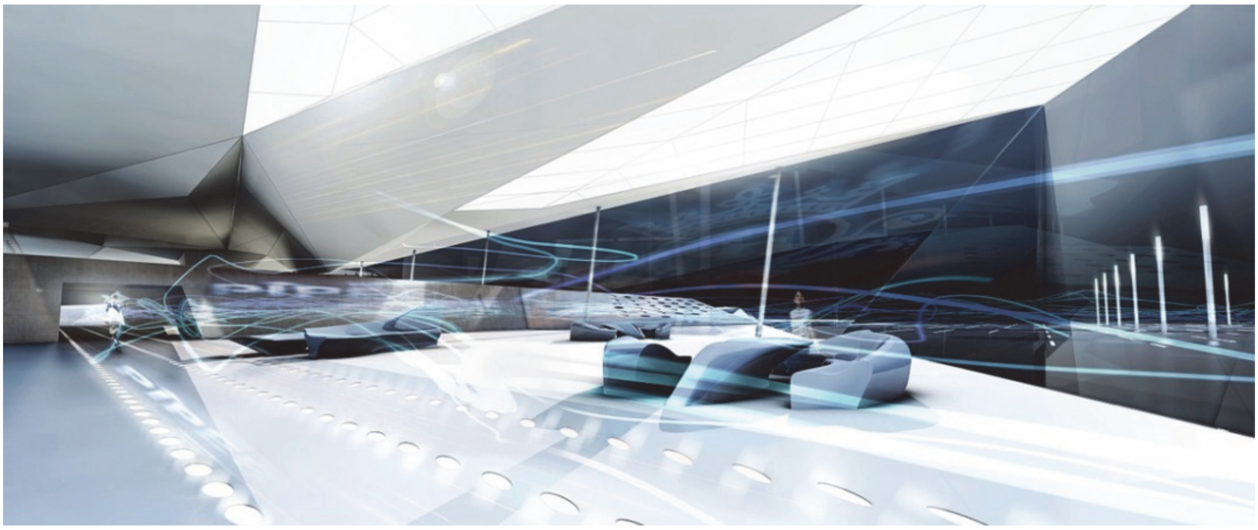


Fig256. *SOM Planner, 2006: rendering by Zaha Hadid architects from their final report*

7.3.2 Associative Use Schema

A more complex representation of the spatial unit was undertaken with CECA student Tim Ireland in 2003. Ireland's MSc project aimed at finding a representation and mechanism to generate a building from inside-out. The organic analogy was based on Frederick Kiesler's conceptual project of the endless house, which Kiesler foresaw as a spatial configuration (of a residential house) correlating to human occupation (Bogner 2001). Kiesler's notion of occupation was not purely a functional definition but mainly a concept of fluid movement called poly-dimensionality, which would give rise to a dynamic looking morphology. The building is therefore depicted as a self-organizing responsive system of the poly-dimensional forms of activity.

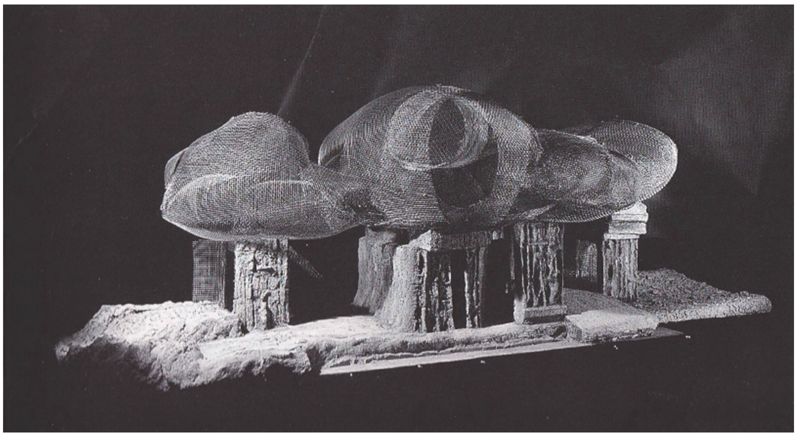


Fig257. *Frederick Kiesler, Endless House, 1924: the maquette of the Endliess House concept shown at MoMA NY, 1958; the spaces were first modelled in chicken wire before layers of plaster were applied*

It was noted that the standard representation of a building in the scientific architectural research field seemed to fail in two ways for Kiesler's concept: in occupation diagrams and spatial theories building layouts were primarily represented as formal room partitions and in two dimensions (Fig258). Formal room partitions employed graph theoretical nodes rendered as circles and the 'occupational' specification was reduced to movement represented via edges between two nodes. This reduction seemed to impede more complex definitions of occupation and it was proposed that to replace a single node by an activity-association matrix. The matrix included the specification of some activity *n* and its preferences to co-occur with other activities. When each activity was defined through associations to other activities, a nested system would result that should self-organize to resolve its partially circular associations. The minimum element, as requested by Hillier and Leaman (1974), would therefore be represented purely via an associative structure.

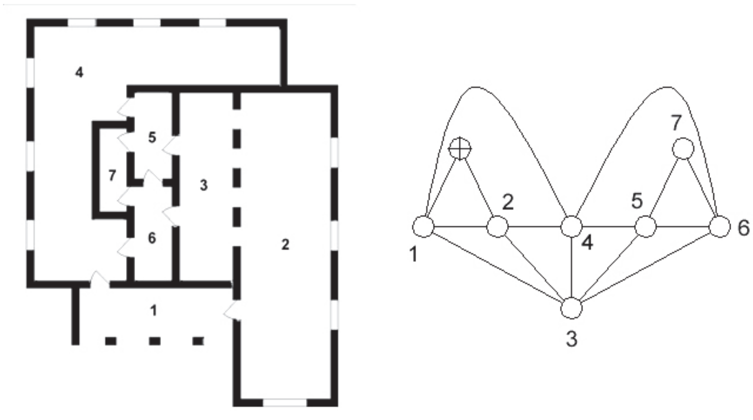


Fig258. *Bill Hillier, justified graph: (left) a building plan and (right) a graph representation where each room as a spatial aggregate is represented by one node (Hillier and Hanson 1984)*

The mechanism identified for attempting this generative representation was the SOS model developed by the author from 1999-2001 (chapter 6.3). A multiple SOS (here called PolySOS) was envisaged where each network would represent an activity of the occupant (Ireland and Derix 2003). The user would select the activities required for the hypothetical house from the GUI at the start of the simulation. No room number, spatial types or other geometric partitions were defined. The implicit space representation of the original SOS also supported Kiesler's argument that space is a three-dimensional personal environment, not an extruded plan (Bogner 2001). Like the SOS, another key concept constituted the autonomous input sampling where the input sample distribution is represented by the sum of all network nodes. But as all networks learn and adjust their structure in 3D, the sample space is consistently changing, establishing a dynamic co-learning environment.

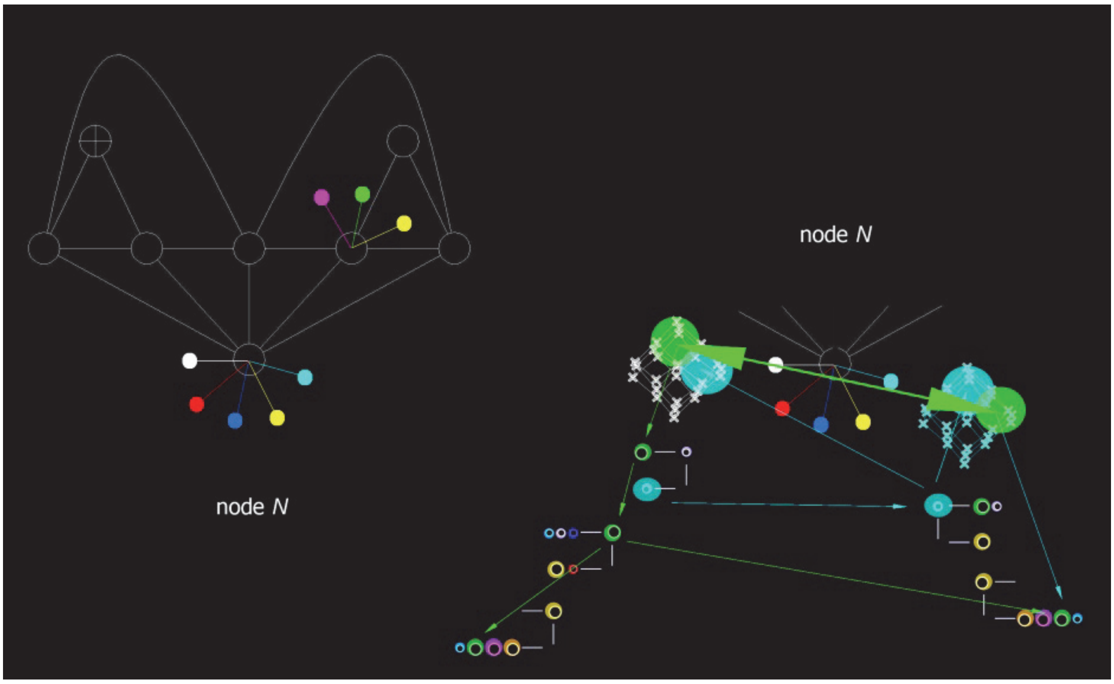


Fig259. *PolySOS, 2003: decomposing each graph node into a collection of activity definitions (left) would translate each graph node into a complex nested relational structure (Ireland and Derix 2003)*

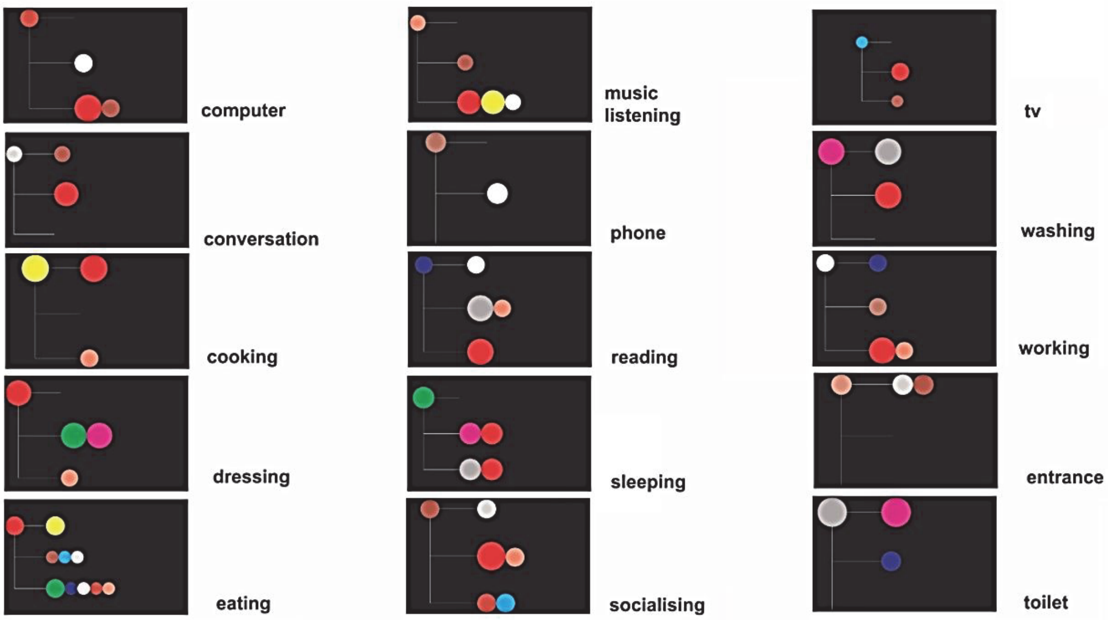


Fig260. *PolySOS, 2003: Venn diagrams portraying the notion of activity associations by a) strengths of association and b) frequency of occurrence; each activity is shown by colour top left node with three levels of association strength shown by the lines; the size of the activity node represents the frequency of occurrence (Ireland and Derix 2003)*

Eight activities could be selected from the GUI where each activity would be specified through four variables: activity type, frequency of use (often/normal/rarely), associations to other activities and space size (number of network nodes). The activity type encoded a frequency of use determining the size of the network by a power function and the learning rate. Associations to other activities are encoded in a table of associations specifying which each activity is connected to. Connections were based on empirical weighting by the observer who could be a designer or any other stakeholder such as the hypothetical occupant himself. All associative connections are weighted by a connection strength that determines the attraction rate between two activities. Hence, there are three types of weighting

- Size of network
- Frequency of activity occurrence
- Association strength

Those are decoded into the learning and topological radii parameters, so that the general feedback weight for a winning neuron is defined through

*learn(n,i) = learn(n,i) * (1 – time/ k) * activity(n).node(i).frequency * activity(n).node(i).association_strength(j)*

where *n* is the activity type and therefore network, *i* the node in the network as a winner, *j* the found input node of another activity network and *k* a constant to balance the monotonically decreasing learning and radius. The learning adjusts the three-dimensional (x,y,z) axis components of each network node, so that a topologically organized configuration and morphology are formed through iterative training. Unlike the standard SOM learning rate, the learning parameters were reset after each generation. Inhibitory feedback for nodes of the same network was applied to separate disparate activity spaces. Winner nodes only organize their own network. Feedback between networks takes place via the association strengths that attract or repel non-winner nodes between networks. A spatial partition or 'room' is therefore represented by an intersecting field of aggregated nodes, not purely by a single network. Perceptive fields are a hybrid of various networks that respond to a mix of usages. If the attraction strength is mutually high across two networks the resulting intersecting space implodes. More balanced strengths between networks produce better distributed and hence more spacious 'rooms'.

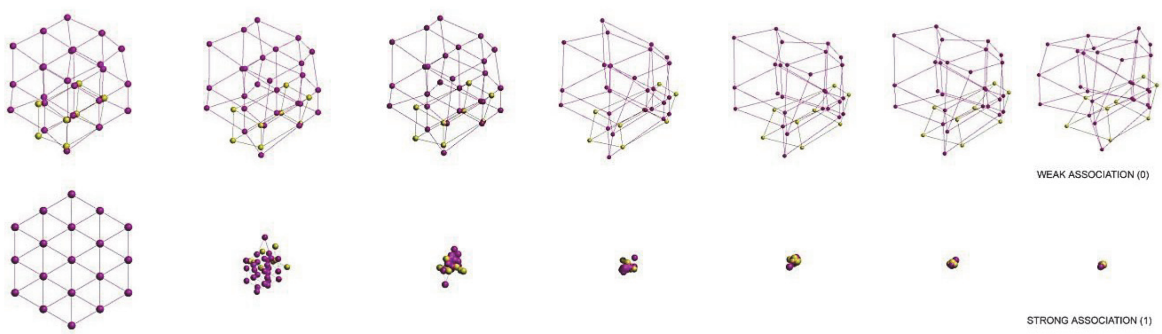


Fig261. *PolySOS, 2003: two networks representing two activity but weighted once with weak strength association (top) and strong assocation (bottom); the weak strength between just two networks leads to some definition of overlapping spaces while a strong association can lead to a simple implosion (Ireland and Derix 2003)*

Space emerges as a co-learned relational field of usage. Although there are spatial units represented as network nodes, each node is only a partial position of a distributed representation of an activity. No formal spatial elements exist, only a relational structure that is mapped dynamically via its associations into three-dimensional space. The observer-designer or observer-occupant weighs activities and their associations from experience and subsequently learns about the effects of his empirical assumptions.

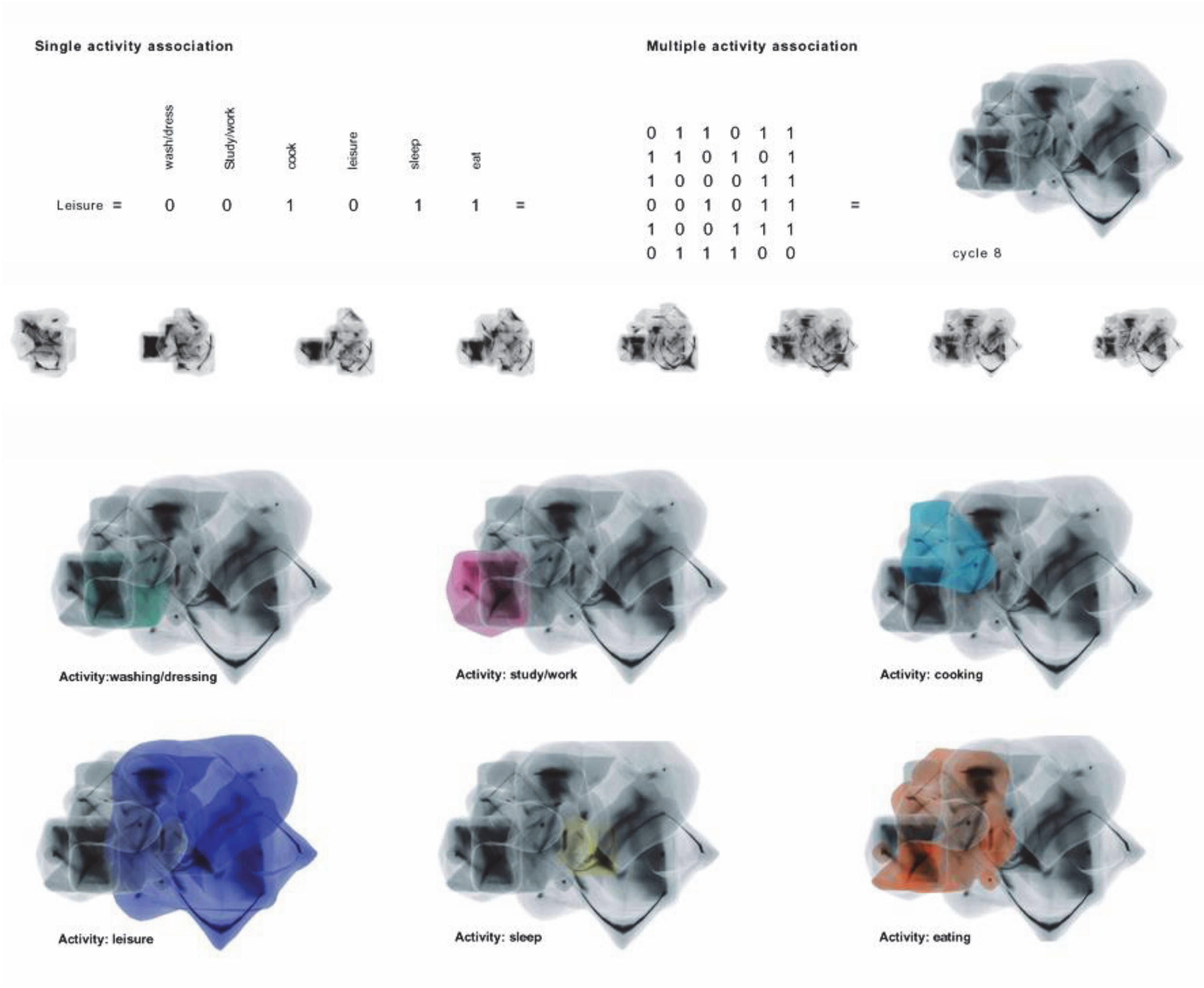


***Fig262**. PolySOS, 2003: interpretation of process and result; (top) the weighting of associations shown as a Boolean matrix, which is the translated into real numbers; (below series) an unfolding by learning of the networks into a configuration, interpreted below into areas of activity spaces that are geometrically interpreted by the implicit surface algorithm described for the SOS; these images were generated for the Future House Competition by Aedas, 2004 and the re-done for the Digital Intuition exhibition by Nous Gallery, curated by the author in 2009*[67]

The PolySOS represents an academic proof-of-concept similar to Kiesler's endless-house concept. Using a single network type as the SOS itself would not be sufficient to analyse and generate spatial configurations by associative feature comparison. The multiple network structure was an important test that led to the research into adaptive topologies discussed in 6.3 and in the following project to a hybrid classification structure (Derix and Jagannath 2014). The PolySOS appeared to approximate Hillier and Leaman's manifold structure in the sense that no distinct formal element was defined, only sets of relations between learned social patterns, spatial relations and design heuristics.

---

[67] http://www.museumofarchitecture.org/exhibitions.html, accessed 15.11.2015

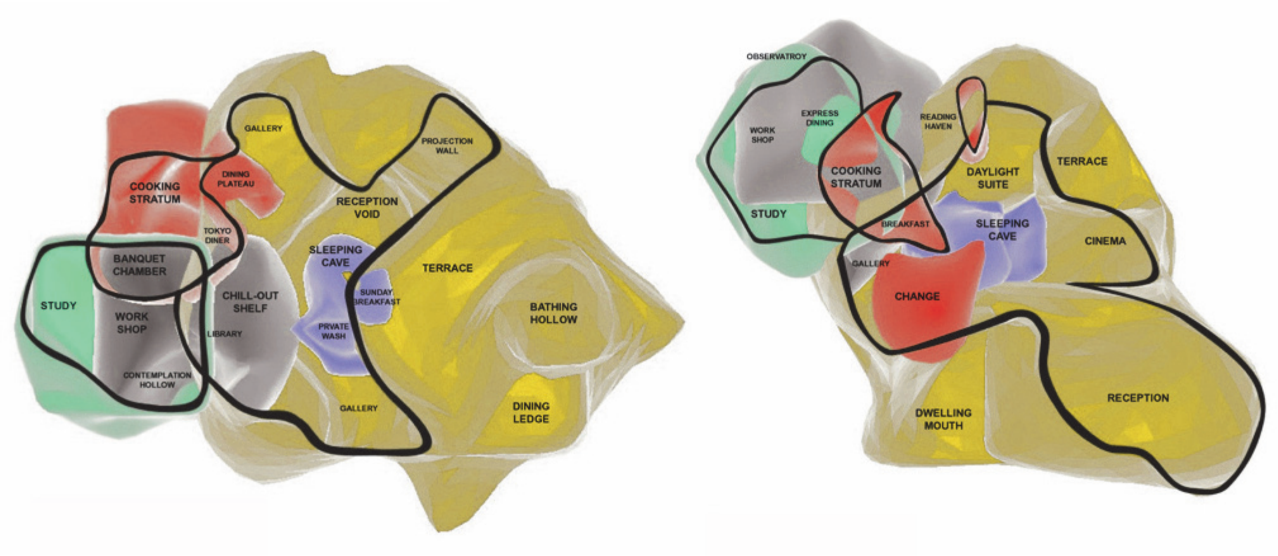


Fig263. *PolySOS, 2003: (top) two sections of an emergent spatial morphology algorithm for the Future House Competition at Aedas 2004, using the implicit surface*

7.3.3 Meta-Cognitive Configuration Of Space

The associative structure of both the SOM Planner and PolySOS consisted of a single layer self-organizing SOM (or multiple SOMs) and an output layer that directly represented the diagrammatic spatial configurations. In 2008 CECA student John Harding devised a multi-stage self-organizing spatial layout system that contained two models of classification and two output formatting models, hence a four stage self-organizing design system. The proof-of-concept system, called Artificial Curator, aimed at a space planning method for an exhibition hall, where exhibitions could be laid out in such a way that qualitative features could be associated with locations (Harding and Derix 2010). This was based on the assumption that people associate place with qualities through memory. Humans cognitively organize space by relations and frames of references (Tversky 2000). Visitors returning to gallery spaces often remember qualities correlating to spaces and create heterogeneous mental maps. The Artificial Curator intended to provide a method by which exhibits could be laid out, so that their qualitative features correspond to places and topological neighbourhoods. Returning visitors could then find it easy to navigate exhibitions by similar qualities across different exhibitions.

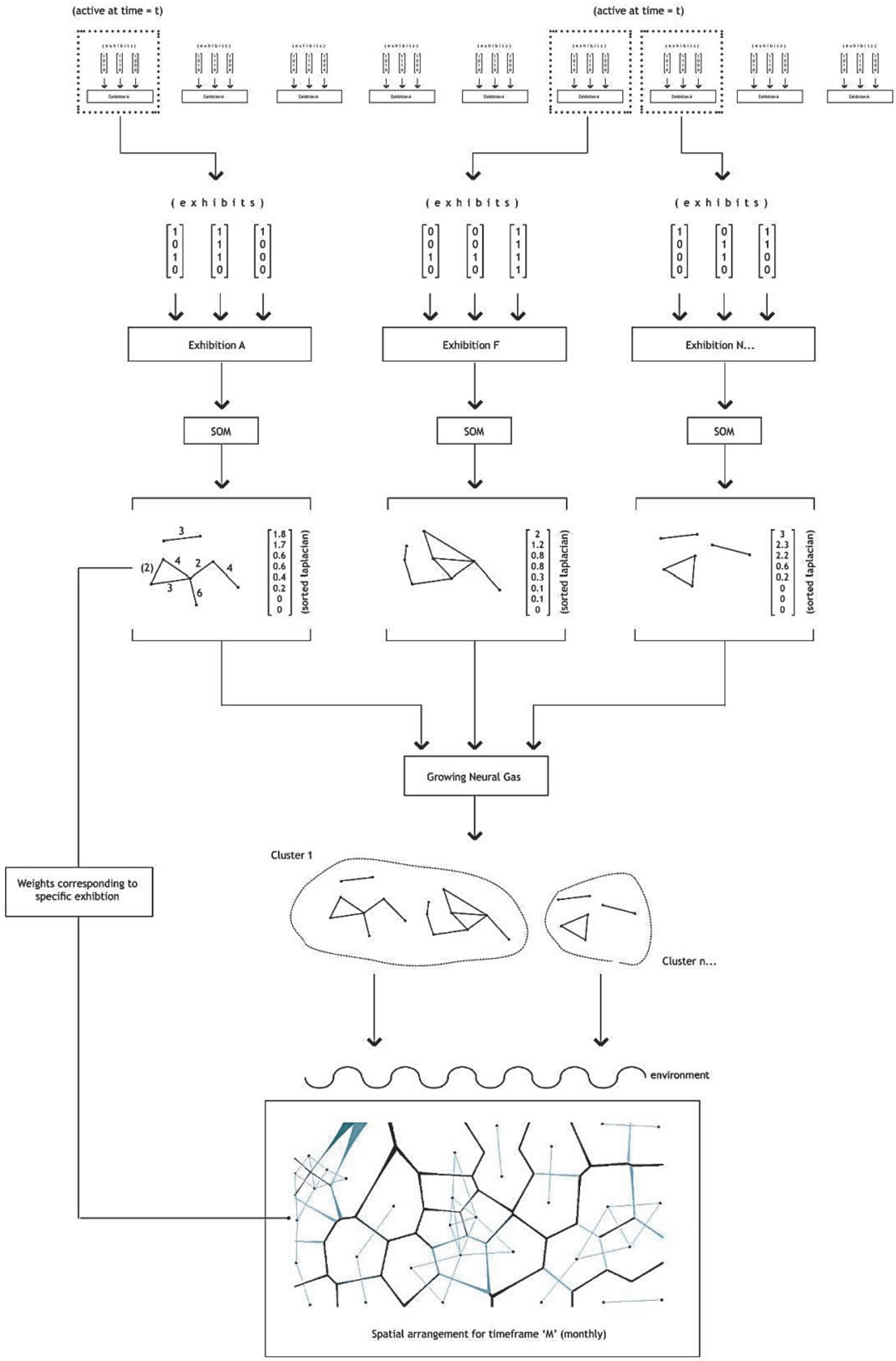


Fig264. *Artificial Curator, 2008: the system's diagram showing four components over two stages: the classification of plan graphs by SOMs (1) and their unique spectra (2); and the generation of similar spatial topology clusters by the GNG (3) and its layout via the spring-system with Voronoi visualization (4)*

The system therefore needed to distinguish exhibit features of exhibition types within an exhibition hall. Qualitative and spatial classification are correlated and mapped into each other to generate layout patterns. Instead of an integrated classification stage that carries out both the difference mapping (learning) and the translation into form (generation), analysis of exhibition qualities by exhibit features and spatial generation based on those qualitative classes are separated. The first classification distributes the spatial topologies of each exhibition into perceptive feature fields and the second classification clusters those feature fields by similarity into new spatial layouts. Topology clusters enable the layouts to be based on cognitive features, creating a kind of 'meta-cognitive' generative map. The two stage mapping also avoids complicated feature formatting and potentially false classifications due to feature overkill. The dimensionality reduction from qualities (30 exhibitions $\times 8^5$ disparate feature combinations) to two dimensional topologies to 16 spatial clusters would most likely be too complex in a single step model.

The Artificial Curator has no pre-defined schema of how to map qualities over time into spatial configurations and equally the observer most likely does not hold an *a priori* concept for this complex task. Qualitative features are to organize space (within a boundary) instead of spatial constraints alone. Whereas the SOM Planner and PolySOS worked on more traditional schemata with spatial and quantitative variables, the Artificial Curator works with spatial and qualitative variables that use the epistemic structure of the ANN system.

The system itself consists of two phases with consecutive four steps (Fig268):

PHASE A - Coding Qualitative Topologies into Plan Graphs
- A1) SOM: generate spatial topologies from features for each exhibition
- A2) Spectral graphs: recode topologies into single input vectors

PHASE B - Generating Spatial Configurations from Clusters
- B3) GNG: find clusters of similar spatial topologies
- B4) Spring system: decode clusters into layout visualized with a weighted Voronoi

A - CODING QUALITATIVE TOPOLOGIES INTO PLAN GRAPHS
Phase A organizes individual exhibitions into topological plan graphs. In step A1, a Kohonen SOM is employed to create associative maps of 30 exhibitions, a map for each exhibition classifying the differences and similarities between the features of exhibits (Fig265). The features are defined through binary values and thus the SOM uses the Hamming distance (as opposed to Euclidean distance or dot-product) to compare the input vectors to the map nodes. While the distribution of samples on a SOM when re-training with identical input samples, the topological configuration of samples on the map remains consistent. Hence, a plan graph is generated from the topological adjacencies that constitute the core learning of the associative network. The plan graph is generated simply by connecting nodes within small radii, so that planarity is guaranteed (Steadman 1983). For the generative layout stage, the Euclidean distances between nodes on the map are calculated, normalized and stored in the graph edges connecting neighbouring nodes. Those generalized

distances help to adjust the eventual layout by metric distances and provide an indication of similarity between exhibits: the nearer = more similar, the farther = more different (purely topological measures would not account for this real-space dimensioning).

	foldable	4 legs	adjustable	armrests	cushioned
Red Blue	0	1	0	1	0
Barcelona	0	1	0	0	1
Butterfly	1	1	1	0	0
Aalto Stool	0	0	0	0	0
Eames	0	0	1	1	1
Bubble	0	0	0	0	1
Deckchair	1	1	1	1	0
Wassily	0	0	0	1	0

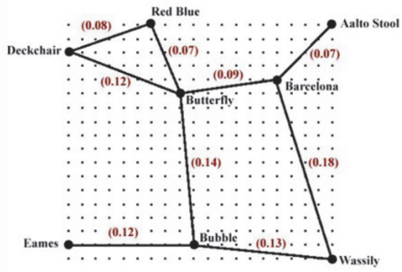


Fig265. *Artificial Curator, 2008: one of the 30 mapped exhibitions shows chairs; (left) the feature break-down of the exhibits into eight chairs with five features and (right) the plan graph interpretation from the SOM topology; the red real numbers along the edges show the normalized Euclidean weights for distances between nodes*

For comparison with other exhibitions, the topological plan graphs need to be generalized into unique vectors, called graph spectra[68]. This is done by a three stage encoding process in step A2, developed by Zhu and Wilson (Zhu and Wilson 2005) and tested on architectural layout classification by Sean Hanna (2007a) at UCL.

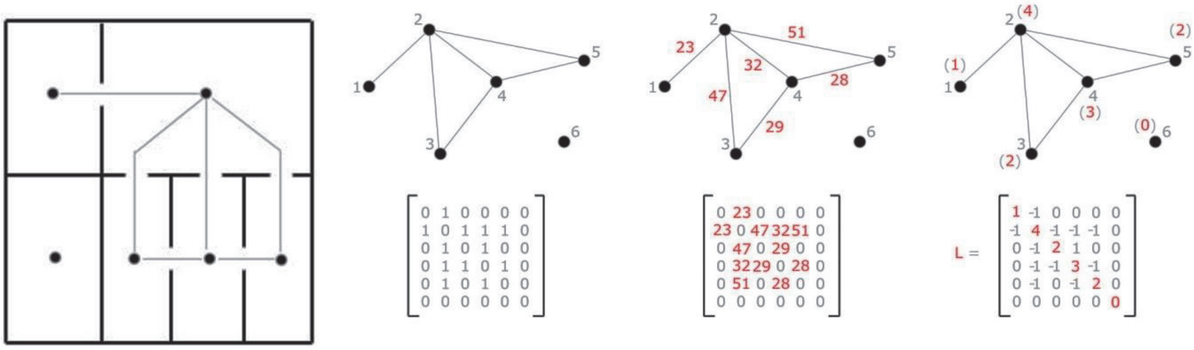


Fig266. *Artificial Curator, 2008: the encoding of the plan graph (right) into adjacency matrix (right-middle), weighted node degree matrix (right middle) and the final Laplacian matrix showing the degree as diagonal and adjacencies as -1 connections*

The plan graphs are encoded into two matrices: an adjacency matrix and a node degree matrix (Fig266). The adjacency matrix represents the connectivity between nodes in the plan graph symmetrically through binary values where 1 = connection. The Laplacian matrix encodes the node degrees, meaning the number of connecting edges into a node that are filled into the diagonal of the matrix. The Laplacian matrix from which the graph spectra are generated, is the difference from the adjacency and node degree matrix (Fig267): Laplacian = Degree – Adjacency. The Laplacian matrix already indicates some interesting graph features such as the number of sub-graphs or separate components in a graph. The spectrum for each plan graph is produced by calculating the *eigenvalues* of the Laplacian matrix, which are then sorted by size in the resulting vector. The spectrum contains properties of a graph like the sub-graphs, connectivity of nodes and spanning trees. Again, sub-graphs can be identified now by the number of zeros at the end of the vector. This is identified through the number of zeros ('0') in the diagonal, with one zero indicating a single graph and two zeros indicating two sub-graphs. The dimensions of the final

[68] Akin to morphological skeletal discussed in section 6.2, where the medial axis diagram reduces a geometric layout into a topological graph as a unique but generalized representation of that space.

vector for classification in the next stage will change as the zeros are not included. Hence, an adaptive topological associative network is required to find clusters of varying dimensionality.

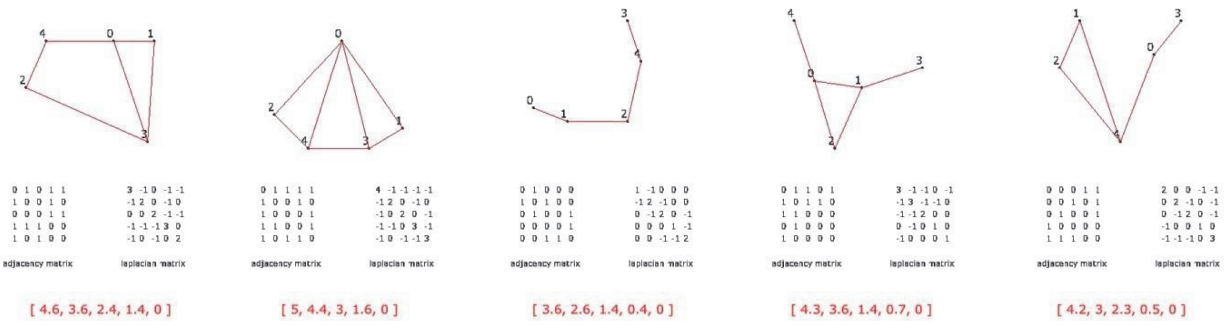


Fig267*. Artificial Curator, 2008: examples of plan graph types with single or multiple cycles or trees; (below middle) the two matrices and (bottom) the resulting graph spectra*

B - GENERATING SPATIAL CONFIGURATIONS FROM CLUSTERS

A catalogue of 30 spectra has been created of the SOM graphs, representing their unique spatial topologies. In order to understand which exhibits within exhibitions can be shown in corresponding spaces (i.e. sequence of exhibitions laid out so that exhibits can be associated to locations) the spectra must be classified by similarities and their properties associated. A Growing Neural Gas algorithm (GNG) as discussed in chapter 6.3 was chosen and extended to generate the topological association clusters. A GNG was used because unlike required by a SOM, no fixed number of clusters was known *a priori* and the topology needed to be adaptable to dynamically varying input feature dimensionality. The topology of the GNG grows the number of nodes and connections necessary to represent the topology of the input space. Due to its fixed input categories and relational structure (topology between nodes), the SOM represents a classifier for the generalization of a topological distribution. The GNG on the other hand, grows an exact topology (relational structure) not generalizing between features but arriving at a maximum number of clusters. This maximum represents the halting function but the GNG can settle earlier if no additional clusters are distinguished from new input samples.

The maximum set of clusters was 16, which was subject to the dimensional constraint of the size of the exhibition hall. 1500 input signals were presented to the GNG from the input space of all spectra. Fritzke's GNG algorithm (1995) is used, which as discussed in 6.3 keeps all learning parameters constant over time allowing for integration of varying dimensionality of spectra. For the plan layout of the network a repel-attraction physical force model was applied as discussed in 6.2, which was originally proposed for adaptive topologies by Fritzke (1994) for his growing cell structures model. Connected edges between inserted neighbouring nodes repel each other when within a certain Euclidean distance radius but simultaneously attract each other when outside this radius. Unconnected nodes apply no force (Fritzke 1994, p1448) called this a *disc embedding visualization*). Eventually, the 16 clusters were classified providing categories for distinct spatial areas within configurations or whole configurations, depending on the cluster size (each exhibition can contain multiple sub-graphs that can be part of various clusters). The clusters were visually distinguishable by properties such as sparselyconnected graphs and sub-graph components (which would result in porous configurations), highly interconnected graphs with internal cycles (which would result in loops in circulation), single tree graph configurations and many similar graphs with two components.

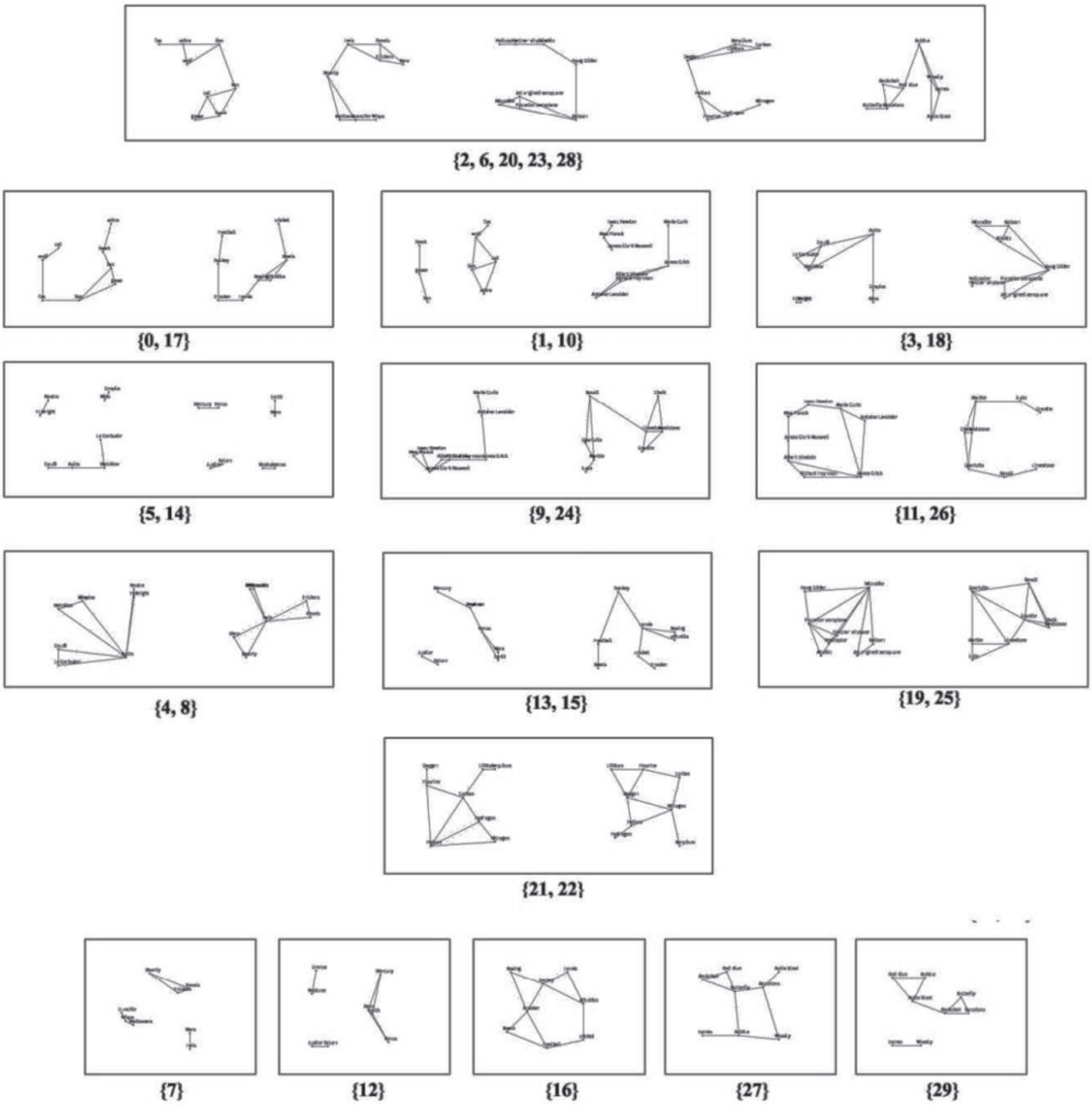


Fig268*. Artificial Curator, 2008: the GNG produced 16 clusters of similar plan graphs from their 30 spectra; many are so different from others they form their own cluster (bottom row)*

Finally, clusters of similar spatial topologies based on similar qualitative features needed to be laid-out into the exhibition hall. The layout of each exhibition is determined by similarities between exhibitions identifying the number of similar clusters within each. This can also be seen as a logistical problem: if sub-spaces between exhibitions are similar in allocation then less effort is required to change exhibition set ups.

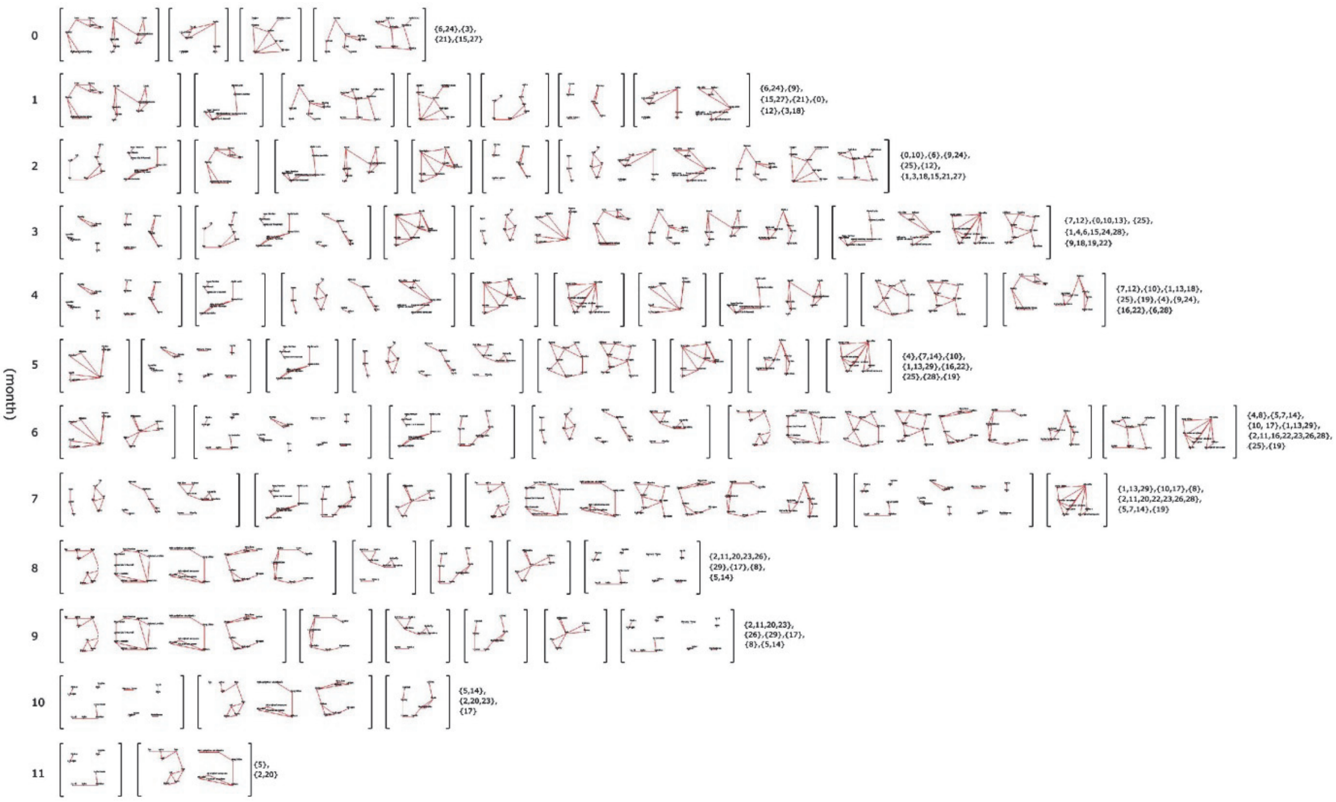


Fig269. *Artificial Curator, 2008: initially during Harding's MSc, the idea was to generate 12 scheduled exhibitions; this figure shows the 12 exhibitions containing different number of clusters; hence, exhibitions could be arranged in sequences of similar number and types of clusters*

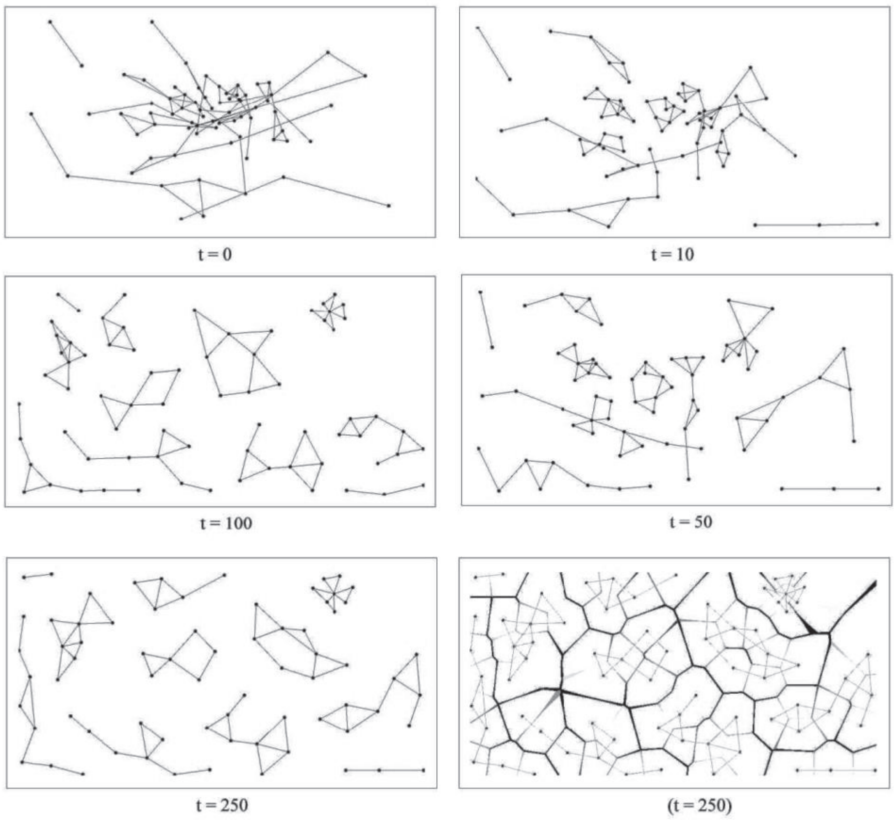


Fig270. *Artificial Curator, 2008: the spring system lays out an exhibition by repelling all nodes, keeping connected clusters together and adjusting distances between exhibits by their plan graph Euclidean weights; when settled, a Voronoi diagram is weighted also by the Euclidean weights of the topology to insert partitions*

To lay out the exhibition hall for sequential exhibitions, one spectrum within a cluster needed to be singled out as the template for configuration. Therefore, all spectra in a cluster were averaged from their vector sum and the spectrum with the least Euclidean distance to all others chosen as the template. This template spectrum provided the seed plan graph to be translated into the space of the exhibition hall. If an exhibition consisted of a variety of clusters, a series of plan graph templates was applied simultaneously. A spring-based repulsion algorithm was used to help unfold the plan graphs that the spectra represented where connected edges were represented by springs (see chapter 6.3). The graph iteratively unfolds until all nodes settle. The metric distances between exhibit nodes and separate components within a configuration were determined by the weighted edges originally generated by the SOM and stored in the plan graph edges. A Voronoi diagram is produced from all nodes within the exhibition hall. Where no topological connection exists, a partition is inserted along the Voronoi boundary. Weak connections inserted permeable boundaries (like visual obstructions) and strong connections do not insert any boundary partition, opening a permeable link and combining exhibits into areas of similar qualities.

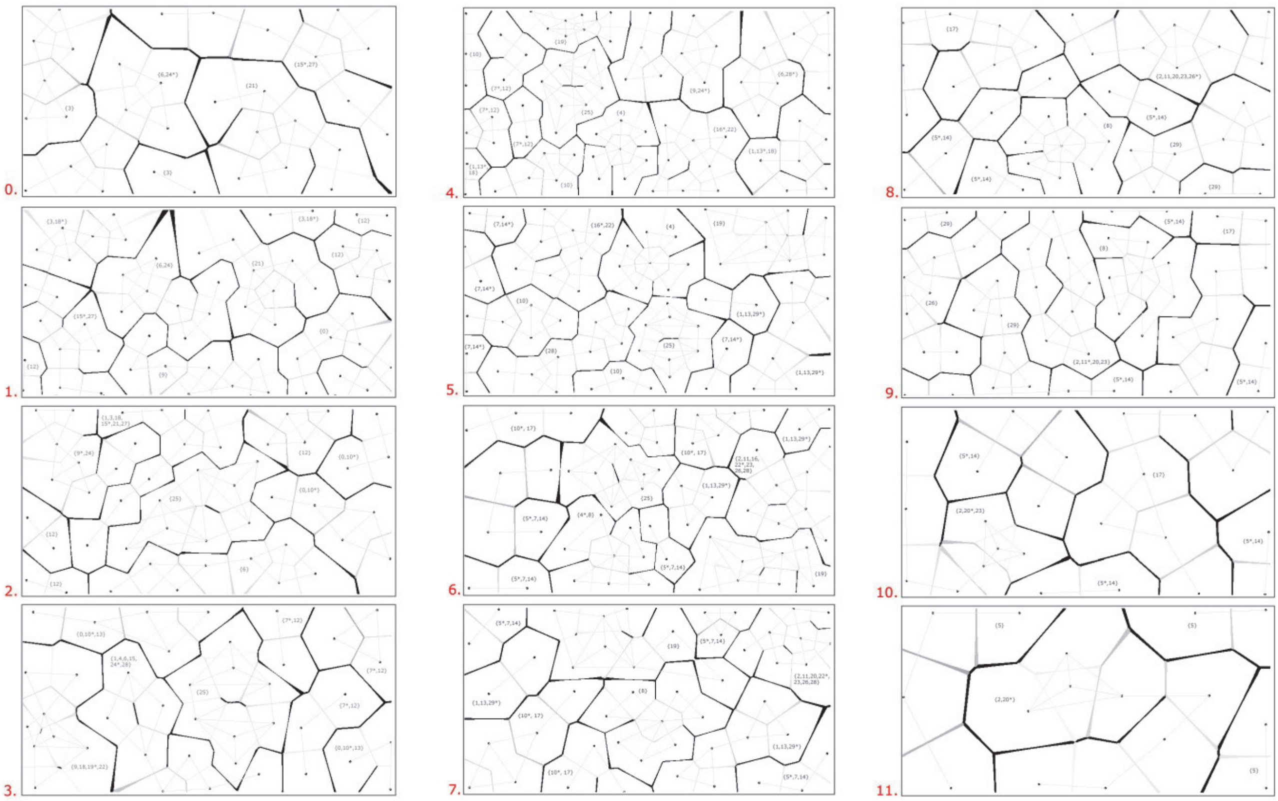


Fig271*. Artificial Curator, 2008: the 12 exhibitions of the original Harding MSc laid-out by the spring system with weighted Voronoi; no particular order was proposed in this image*

CONCLUSIONS FROM THE ARTIFICIAL CURATOR

Like Frazer's electronic models of the Generator, the observer as user and designer is learning from the system rather than teaching it. The input feature samples are not pre-selected by the observer and no schemata exist by which the observer pre-empts the ontology of the samples. The system reflects more the observer *in space* than the designer *of space*, anticipating cognitive behaviour of navigating visitors by empirical associations between space and non-spatial properties. The visitor as observer cognitively emulates the spatial configuration as outer environment and simulates its topological structure through an inner environment. Thus, he encodes a correlation between himself and his environment through an associative bodily (neural) structure, much as proposed in the original German empathy theory of the late 19th century (Schwarzer 1991).

The Artificial Curator also reflects Cruise's distinction (2005) between tactical and strategic cognitive heuristics. Clearly, the observer as visitor does not construct anything physically like Cruise's project heuristics but regenerates an experience, which is simulated by the system. The observer-system conducts comparative analysis by association to the context through tactical decisions, which here is done by the SOM. Then it recreates a spatial construct through a heuristic strategy as done by the GNG. Hence, while two analytical techniques are applied, the overall system workflow produces a generative experience akin to designing spaces.

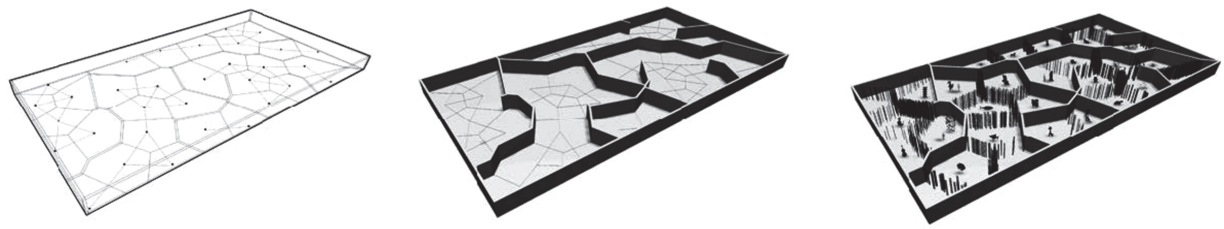


Fig272. *Artificial Curator, 2008: the spatial configuration of one exhibition visualized with the Voronoi (left), inserted solid partitions due to Euclidean weights on topological connections (middle) and the permeable partitions with a place-holder exhibit at the centre of the cells (right)*

7.4 CONCLUSIONS

The chapter discussed three types of models synthesizing generative and analytical heuristics into design computation models: remote observer, situated observer and learning observer. The remote observer applies field performances from analytical models as a global evaluation agency. The situated observer is integrated into the design process by acting locally through interactive changes to the model and thus becomes an internal mediator, situating his evaluation agency through real-time inspection. The learning observer is neither external/ global nor internal/ local but a corresponding designer. Negroponte suggested that design with computers should be like a dialogue rather than the optimization of criteria. While the first models of the remote and situated observer complied with his proposed concept of dialogue where "*the machine would act in 'interrupt' or 'reply' to its partners"* (Negroponte 1970, p39), models of the learning observer extend this analogy by allowing the machine to generate topics of conversations instead of mere responses. In this situation, both conversants maintain epistemological autonomy.

7.4.1 Co-Evolving Intentions

As such, the observer's design intentions and KPIs for evaluation are shifting from established industrial schemata applying mainly quantitative targets to human-centric schemata using behavioural performances and eventually to correlational schema using cognitive qualities. Established schemata are considered to be epistemically closed as the knowledge to be produced is set by the observer. Different levels of abstractions are used for representations of schemata from discrete values for external targets of quantitative schemata (first order), to aggregated diagrams for internal performances of user-centric schemata (second order) and finally numerical associations processing discrete values and aggregate diagrams for cognitive schemata (third order). As intentions and KPIs of known external schemata (i.e. about people and spaces outside the observer) are replaced by evolving internal cognitive correlations to form new schemata, the observer generates concepts and their performance evaluations simultaneously with the model. Instead of an observer-guided evolution, a co-evolution of intentions occurs through empirical dialogue, akin to Schön's definition (1983, p185) of the *generative metaphor* that evolves through reflection.

7.4.2 Model Structures

If epistemically closed[69] generative heuristics as in section 5.1 are coupled with spatial analysis which are partially closed then the question arises what balance a synthetic model achieves between generative and analytical drivers and how epistemologically open the models are. The question of knowledge belonging to the model rather than observer has been addressed in the previous section where a shift from knowing to learning observer takes place.

Generative heuristic of the remote observer are simply constrained by the KPIs in form of quantitative values determines by analytical heuristics of industrial schemata. A generative algorithm produces configurations, and a single heuristic is simulated. The situated observer on the other mediates two distinct heuristics: the performance analysis and the generative heuristic. In the Euston Crossing model for instance, there is a generative model using two algorithms - Dijkstra and Gaussian distributions - and an analytical model using two visibility algorithms – VPTA and ray-tracing. The observer's interaction counts as another mediated heuristic, i.e. the agency of the observer himself. Hence, models of the situated observer are structured into three components, where the observer mediates associations as a local agency. This represents a qualitative shift from the mediated generative model structures of chapter 5.1 where for example, a CA transition function negotiated neighbouring states within an algorithm. The role of the CA's transition function is taken over by the observer to mediate between analytical and generative algorithms to search the design space. Analytical and generative heuristics become equivalent.

Models of the learning observer vary in structure but share the principle that what is usually perceived as an analytical model – the ANNs as pattern recognition or data mining algorithm – also provides the generative agency. The associative process searches and generates schemata and aligns with the observer's cognitive heuristic of categorizing empirical phenomena. The final model, Artificial Curator, eventually unwinds the single algorithm model into a computationally unconventional inverse sequence of analysis first and generation second. This should not be confused with Alexander's approach (1964) of the *good fit*, but instead of following the standard generate-and-test principle, a concept is tested for its empirical assumptions and then phenotypes generated, evoking Hillier's *conjecture-testing* mechanism (see

[69] By epistemologically-closed the knowledge production process is meant. In most meta-heuristic algorithms, generated knowledge is dependent on the algorithm, not so much the observer-designer.

2.7.3). Similarly, this approach was described through tactical and strategic heuristics. Instead of generating solutions constrained by schemata, the learning observer generates schemata that produce concepts which constrain future solutions. The observer sits *next to* or gets absorbed in the algorithmic system and generates narratives.

7.4.3 Computational Teleology | Halting Function

The status of the observer also impacts on the teleology of the process. While the remote observer's target aspirations introduce explicit cost functions that the model optimizes towards (teleological process), the situated and learning observer is not setting goal states for the process to attain (non-teleological process). A teleological process is characteristic of mainstream parametric models (and also KbD), where the goal state (schema) is reverse-engineered into a fixed ontology and its dependencies and a solution path is defined that procedurally attains the goal state. The goal state is specified mostly via external quantities like the fitness function of a GA and the model repeatedly executes the same algorithm to approach the desired state. When the goal state satisfices target requirements, the *halting function* terminates the execution by not entering another generation.

Non-teleological models on the other hand, do not apply generations or goal states. It could be said that such models are searching for state groups that allow for reverse conclusions about the definition of the problem (problem-worrying). When observer interaction mediates association to generate states, events become the driver of a loosely associated set of algorithms. The model is always live such as Frei Otto's physical models of material computing. Similarly, learning observer models have no attainment state but simply a state of 'boredom', meaning no more differences can be distinguished from samples of the input space. Clearly, this is not strictly true for the generic SOM whose learning parameters are decaying but for the SOS, GNG and more complex systems like the Artificial Curator it holds that any change in the feature space keeps the system learning. States as solutions are deferred when it comes to autonomous second-order cybernetic systems for which von Foerster (1984, p295) said: "*a change of sensation = change of shape*".

A new correlation between model and schema type emerges: non-teleological models are used to approximate spatial configurations performing well for user-behaviours and cognitive affordances. Teleological models with goal states are less suited for human-centric performances, because explicit *a priori* cost functions are much harder to define. In other words, where human-centric aspiration states are sought after, undirected models of situated and learning observers are preferable

8 FIELD ORGANIZATIONS | HUMAN-CENTRIC SYSTEMS

This last chapter of case studies aims to discuss a type of model that mediates the observer with other components of the design system. No hierarchy is meant to be detectable, creating design systems where all parts are autonomous as much as the process as a whole. This type of computational design model is new, albeit not theoretically but in its attempted practical application and its consequences.[70] However, the two first case studies stem from academia where proof-of-concept models were developed to test the complete assimilation of the observer into a simulated spatial planning system. The last section of this chapter then introduces the translation and its effects of such a concept into practice, presenting a framework approach that is new to practice as much as academia.

As stated in the introduction to the last chapter, the models of chapter seven still distinguish between the observer as a global design lead, setting targets or schemata. Models in this chapter attempt a complete assimilation of the observer as just another local component of a system, integrating him into the field as it were. Hence, the observer is simultaneously constraining behaviours of a spatial configuration as much as being constrained by them. He is subsumed in both the aggregating objects in the field and the affordances of the field. In the models of the previous chapter, the observer was partial, reflecting his positioned agency within the system. When Allen posited that "*field conditions are bottom-up phenomena, defined not by overarching geometrical schemas but by intricate local connections"* (Allen 2008, p218)*,* the question arises how that works? His example of the Mosque in Cordoba as a prototypical field condition shows that he implies graphical patterns and their composition rules (Allen 1997, p27). But he does not provide any clue as to who perceives and acts on those patterns. Allen regards those graphic patterns simply as propagations of difference from formal rules, meaning some generative algorithm (analogue or digital). In other words, his concept of field focuses solely on the single dimension of visual appearance. If instead, they are not simply meant to be graphical pattern but spatial configurations for occupation then implicit dynamics require inclusion into a multi-dimensional algorithmic system[71].

Not all components of a system might be purely bottom-up processes but turn into top-down effects, even though they are locally acting. The models of the previous chapter had this explicit character where the observer would act locally but as global driver. Models in this chapter are instead reminiscent of Herman Haken's *Synergetics* theory (2004) where local dynamics between simple elements can generate attractors that organize the field temporarily into global structures. Haken termed

---

[70] As Stan Allen pointed out:" *The theoretical model proposed here anticipates its own irrelevance in the face of the realities of practice. These are working concepts, derived from experimentation in contact with the real. Field conditions intentionally mix high theory with low practices. The working assumption here is that architectural theory does not arise in a vacuum, but always in a complex dialogue with on-going practice.*" (Allen 1997, p26)

[71] Allen only hinted at this omission by mentioning that *"Finally, a complete examination of the implications of field conditions in architecture would necessarily reflect the complex and dynamic behaviours of architecture's users, and speculate on new methodologies to model program and space."* (Allen 1997, p27)

that kind of global-local coupling the *enslavement principle*, since some local areas of action control the system for some duration. But the parts of those areas are not programmed to be controlling or distinguishable via their ontology. This hierarchical and ontological equality is also discussed in Bruno Latour's *actor-network theory* (ANT) (1999). Latour proposed that a model of society should not differentiate between types of actors and that actors do not determining the structure of the network. Actors simply take decisions that connect them to other actors, which in turn modify network properties[72]. He makes no distinction between figure (actors) and ground (context) but proposes that narratives from semiotic configurations only exist as weighted associations. In other words, he proposes a field of heterogeneous actions that generates global configurations, controlled non-teleologically by local areas (high connectivity and other network measures). The observer in such models as Haken's Syngergetics or Latour's ANT becomes a part of the connected event structure. Agency is not applied from outside but happens uncontrolled locally. The observer thus becomes part of the manifold and systems created by an observer-designer that represent a *behaviour modifier* as Hillier and Leaman argued (1974, p8): "*Buildings mediate two different kinds of relationship between man and nature and between man and man. In mediating relations between himself and nature, man builds a climate modifier. In mediating relations between himself and other men, man builds a behaviour modifier"*.

Why do two non-architectural models – Synergetics and ANT - serve as analogies for this perceived new type of human-centric computational design system? Chapter two introduced the key paradigms that inform this thesis. But all paradigms fell short of implementing their theses on live projects and especially as theoretical and technical combinations. The thinking of the New Epistemologists is reflected in Allen's writing who gives their models a new frame – the concept of field conditions, where only purely bottom-up autonomic systems exist. Mathematical representations of the LUBFS and knowledge representations of KbD provide procedural models but no field conditions. The work of Space Syntax and connectionism dedicated themselves to human-centric spatial analysis, ignoring design generation (or in the case of Coyne, supervised generation without analysis). All attempt to *solve* their design paradigm via single-component algorithmic models. Synergetics and ANT break the mould by understanding that an applied model of science and practice requires a blend when implemented. Paradigmatic axioms are discarded in favour of applicable theories.

Structure

Apart from the role and agency of the observer, a key difference in the case study models discussed here is their structure. Already the Artificial Curator of 7.3 introduced a multiple component model approach, akin to Coyne's proposal of a

---

[72] Latour labours the non-spatial quality of networks to an untenable degree by negating all spatial qualities on social tissue, attempting to liberate social theory from geography: "*The notion of network, in its barest topological outline, allows us already to reshuffle spatial metaphors that have rendered the study of society-nature so difficult: close and far, up and down, local and global, inside and outside. They are replaced by associations and connections." (*Latour 1999, p373)

spatial synthesis planning organization (Fig277). A multiple component model when not sequentially aligned becomes a system that dissolved the boundaries between generating & testing, similar to models of 7.2. The key discussion in this chapter revolves around the question of how to associate apparently autonomous process-components of a system without a strict hierarchy into an observer-objects-field system where human-centric properties, algorithmic and analogue heuristics mediate.

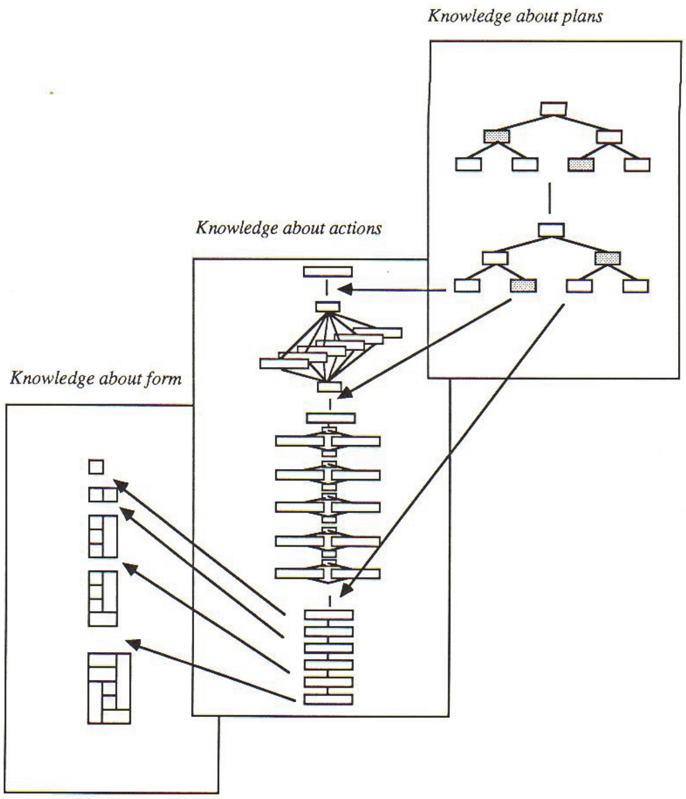


Fig273. *Richard Coyne, 1989: A directed yet associative structure of knowledge modules for a knowledge-based design system; following this publication Coyne abandonded the directed association for connectionism but kept the multi-modular system structure (Coyne et al. 1989)*

Only two case studies are introduced as self-contained projects, both from academia. The third and concluding section discusses the ultimate consequence of an ill-defined computational design system as an open framework. The three model-systems are arranged so that they reflect the narrative of chapter five, six and seven, where the first section integrates the observer as an agent, the second as a participant and the third as an equivalent conversant.

8.1 CONSENSUAL SEARCH | PARTS-FIELD MEDIATION

"When two or more organisms interact recursively as structurally plastic systems, […] the result is mutual ontogenic structural coupling. […] For an observer, the domain of interactions specified through such ontogenic structural coupling appears as a network of sequences of mutually triggering interlocked conducts. […] The various conduct or behaviors involved are both arbitrary and contextual. The behaviors are arbitrary because they can have any form as long as they operate as triggering perturbations in the interactions; they are contextual because their

participation in the interlocked interactions of the domain is defined only with respect to the interactions that constitute the domain. […] I shall call the domain of interlocked conducts a 'consensual domain'." (Maturana 1978, p47)

At CECA, Humberto Maturana's concept of autopoiesis formed a theoretical basis for the R&D of computational design (Derix and Thum 2000).[73] The concept of *consensual domain* proposed by Maturana identified a shared domain of states – knowledge, form, semiotic etc - between two or more interacting systems when their processes mutually adapt to changes in the field, triggered by one of the participating systems. In the above quote, Maturana suggests that this domain is perceived by an observer as a single system, rather than a coupling process between many systems. The observer learns from the dynamics of a model either behaviourally by interaction if he is involved, or cognitively by establishing a mental correlation of what consensus is perceived as a constructive state of a system (or both). The system in that case remains epistemologically autonomous while its structure could have been developed by the observer.

Many examples of the concept of consensual domain through structural coupling had been provided by New Epistemologists but they tended to resort to literal implementations like Coates' co-evolutionary models (Coates et al. 2001), agent-based models like swarms (Miranda and Coates 2000) or even the SOS by the author. Those models use a single algorithmic model as the driver of the system which evaluates and instructs actions. Further, no behavioural or cognitive heuristics relating to an architectural design process were aimed for, producing reductive morphologies devoid of interpretation for human use. Ideally, a consensual domain is generated from parallel processes that interact within a field where their states serve as mutual conditions. This field is what an observer perceives as the state of a single system (with the configuration not visible). Latour and Maturana agree when proposing that no figure-ground or contextual processes are identified, simply a single *network of consequences* (Latour 1999).

8.1.1 SYNERGETIC PLANNING

In 2004, the author was invited by Professor Lidia Diappi of the Milan Polytechnic to develop a model to simulate the assumptions behind the New Urbanism theory (Katz 1993). This model was intended for students of the Laboratorio di Sintesi Finale (final diploma studio) at the Department of Architecture and Planning (DIAP) to understand rule-based correlations between regulation and morphology. New Urbanism was proposed as a case study because it contains a series of constraints for urban planning that are meant to enhance the sustainability of place, mainly via walkable neighbourhoods. The proposal to regulate for walkable neighbourhoods and correlating morphological scale is not new and has been very well detailed into design guidance in Europe by initiatives such as the Urban Task Force's Towards an Urban Renaissance (Rogers 1999).

[73] Paul Coates used the concept of *consensual construction* and *illustrations of consensus* as a description for the mechanism of epistemologically autonomous systems and their states (Coates 2010, p12).

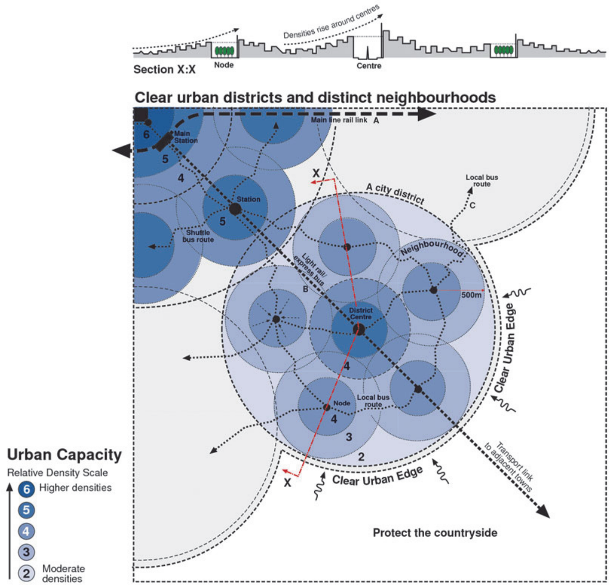


Fig274. *Urban Task Force, 1999: urban morphology (scale, density and district) according to accessibility*

Few computational models for generative urban planning existed in 2004 and Diappi wanted her students to understand the dynamics of urban form through a simulation model. Paul Coates' alpha syntax model initially developed for the publication Social Logic of Space (Hillier and Hanson 1984) and later applied to a variety of student projects at CECA (Coates and Derix 2007) provided a precedent for a generative systemic planning simulation. The alpha syntax model written in StarLogo at CECA in 1999 consisted of three types of cells – house (solid), road (permeable) and garden (semi-permeable) – that were associated via state transition rules in a CA. The output equilibria states produced patterns that structurally and visually matched – in this case - the Arabic city of Sana with its public gardens. Like the original Space Syntax theory (Hillier et al. 1976), only permeable and solid cells were assumed necessary to generate large scale patterns of urban structure.

Fig275. *Paul Coates, 1999: the Sanaa model implementing the 'alpha syntax' for a CECA student; (right) seeded north-south and east-west axes and a series showing growth stages with white cells as street, red as solid and green as gardens (Coates and Derix 2007)*

Other generative models of urban planning that existed at the time such as the Kaisersroth village generation model at the CAAD chair of ETH Zurich (Braach 2014) and the beginnings of the Procedural CityEngine also of ETH Zurich[74]. Both however employed a formal algorithm to generate an *a priori* road pattern (the former from a Voronoi diagram and the latter from an L-system) without accounting for behavioural use or associations to any other aspect of urban form. Schematic

[74] bought by ESRI: www.esri.com/software/cityengine, accessed 10.12.2014

massing was sequentially filled into the generated network as hierarchies of geometry, representing essentially some of the first parametric urban modellers.

The urban planning model at DIAP (SynergeticUrbanism = SynUrb) on the other hand aimed at a synchronous multi-process system where heterogeneous algorithms would generate an urban structure through self-regulation. Each algorithm represented some behavioural or cognitive agency. It was unknown ahead of time which system would dominate the field states at any given time during runtime, aiming to represent an analogous model to Haken's Synergetics.

SynUrb represented a growth model at the street scale, which neither Kaisersroth nor CityEngine provided. Local morphological effects were to be taken into account by simulated abstractions of occupants to weight global configurational effects, mediating tactical with strategic decisions. The model could be used as an in-fill process within an existing context (regeneration) or in isolation for new development.

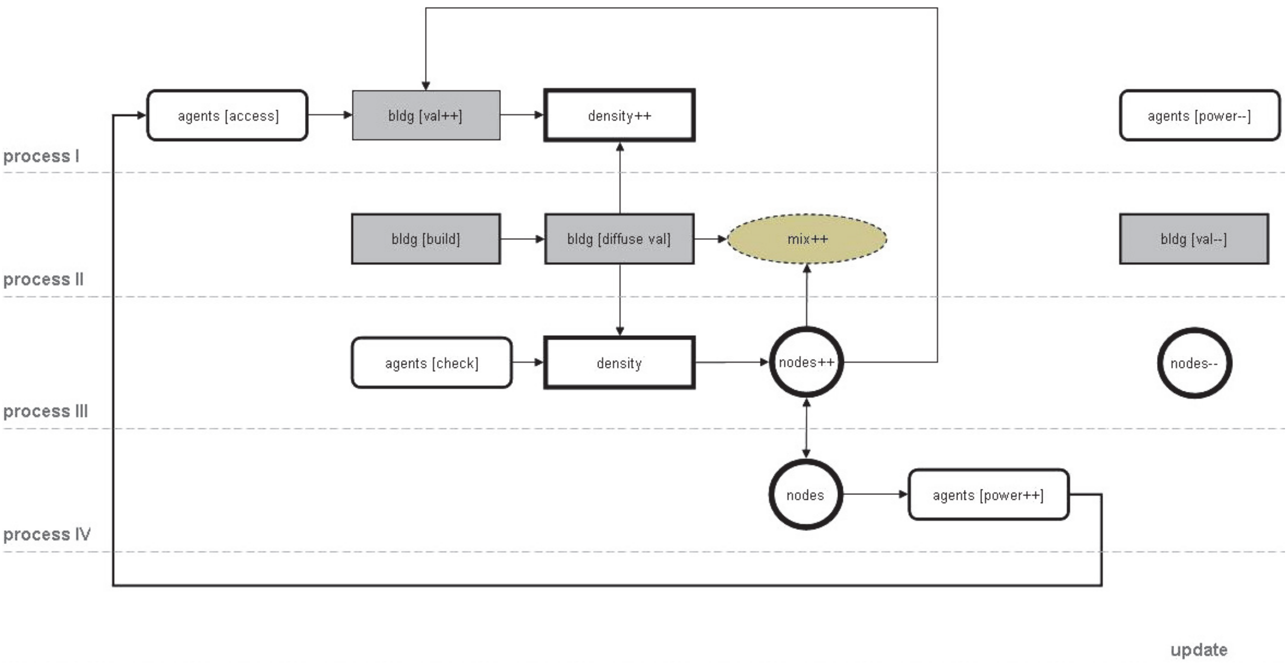


Fig276. *SynUrb, 2004: system diagram, showing three synchronous system processes and their associative feedback*

SynUrb consisted of three algorithmic components:

- Structure multi-agents for movement and accessibility
- Mix cellular automaton for massing scale
- Density graph-based diagram for transport nodes and density

All variables to constrain the min/max extents of processes concerning the structure, mix and density were set in a GUI at the start of the simulation. No further interaction was provided after the initial weighting of variables, turning the user into a remote observer. The associative weighting between the algorithmic performance and configuration states was done by the author writing the model code, first using a NetLogo prototype to test systemic dynamics and secondly using VB for AutoCAD to generate morphological results.

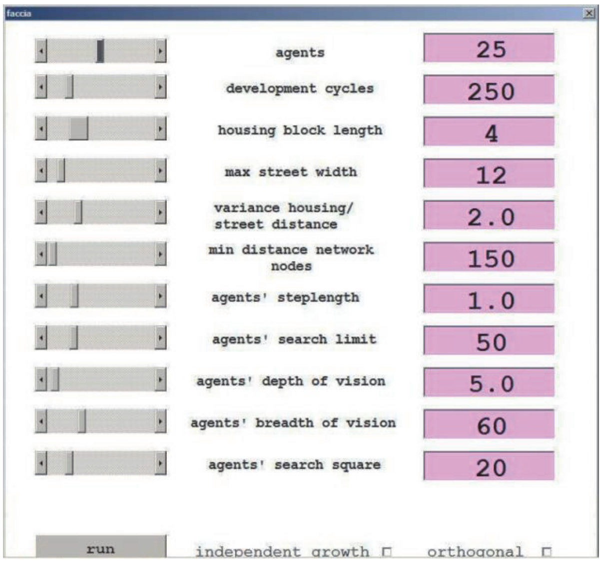


Fig277. *SynUrb, 2004: GUI showing three strategic variables (agents, cycles and independent growth = green field) and ten tactical variables*

STRUCTURE THROUGH MOVEMENT

The question of how a growth process begins uses the assumption that where occupants gather some shelter must be provided, following Aldo Rossi's argument of the *genius loci* generating place (Rossi 1966). Other valid ways could have been used for seeding an initial unit manually or using some contextual stimuli ('Independent Growth' tick box selection in GUI: agents find a land-use unit from which growth rules are triggered). Abstracted pedestrians are implementing a multi-agent algorithm that uses a similar extended swarming method as discussed in 6.1: agents use the flocking method to implement a social aspect of behavior, extended by a leader-following addition, obstacle-avoidance, exhaustion and particularly a scale-dependent direction setting. All methods relate to an awareness of context with the scale-dependent navigation as cognitive sensor of the morphology.

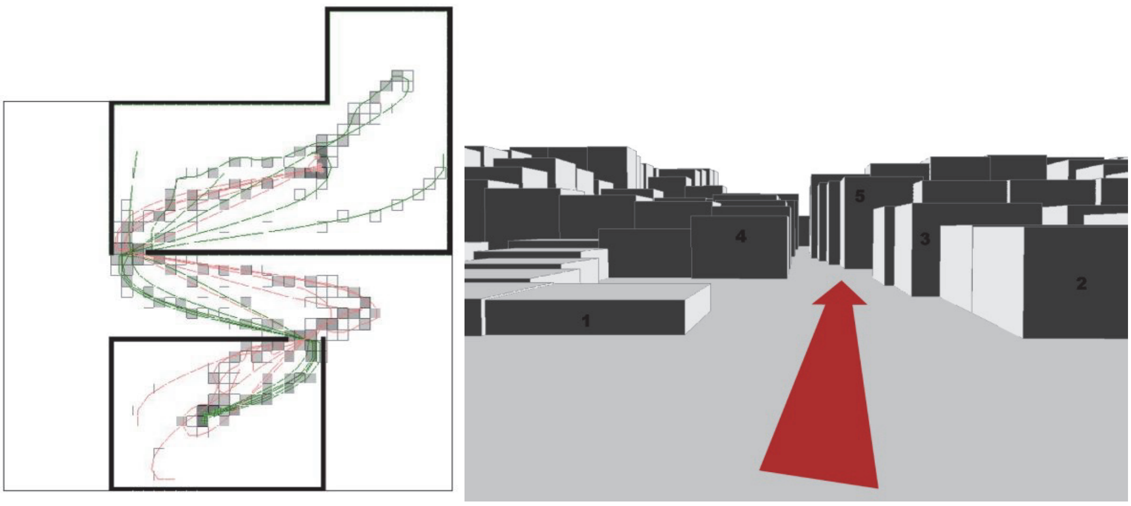


Fig278. *SynUrb, 2004: (left) the first People Movement simulation built by the author in 2005 (see 5.2.1) and an agent's perception of scale, sorting building units by heights in the FOV*

The agency of the multi-agent algorithm is a hybrid between analysis and generation: they try to way-find towards public transport access nodes and by searching for these nodes, they use the morphology as an aide rewarding successful structures. If agents can access a transport node the building units they pass are rated by increasing their value. The passing of building units is however dependent on their chosen route, which is calculated by scale perception:

- search field-of-view for access node
- If seen: orientate your direction towards node, check obstructions and move
- If not seen: create list of building units in sight and sort by height
- set direction towards highest building and check obstructions

The assumption is made that people use massing density (scale and compactness)[75] as perceptual aide when searching for public transport, because density and scale are assumed indicate levels of activity and infrastructure (Rogers 2004; Jabareen 2006) (see PTALs Fig136) . Agents are essentially hill-climbing by building unit scales. London urban planners Tibbalds confirmed this assumption as a planning heuristic: transport access points are often allocated and associated to widening of streets or a negative street-aspect ratio, i.e. buildings are proportionally too high. Because agents simultaneously check for obstacle-avoidance, the chosen direction would most likely be some form of open corridor between building units, semantically defined as 'street'. The morphological property of 'street-ness' is supported by the agents' flocking algorithm. Behaviours such as alignment and cohesion produce streams that when applied in concert with leader-following ('who sees a transport node') produces footfall patterns along successful corridors.

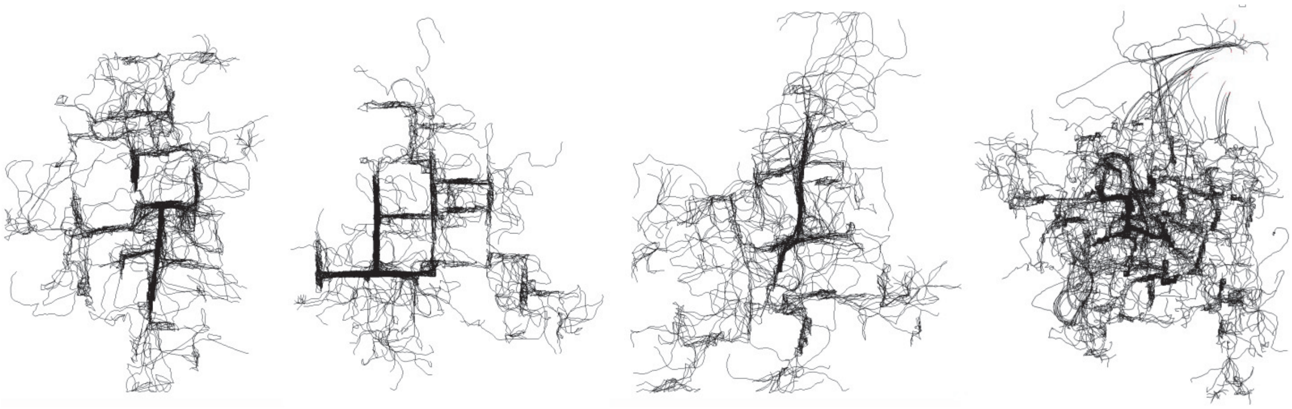


Fig279. *SynUrb, 2004: four states with different morphological weighting of urban structure, generated from the multi-agent algorithm's history of walks (the ontogeny of a system); corridors emerge from footfall that in turn constrain the movement of agents and reinforce the morphology*

Agents are assigned levels of exhaustion (called *power* in the system diagram or *search limit* in the GUI) encoded as number of steps. The perceptive scale search for a transport node is initiated when the level of steps is exhausted, replenishing the level when a node is found. The power level provides the probability through a roulette wheel function by which a building unit adjacent to the walking agent is rewarded. If the random value falls below a threshold of exhaustion, a building is awarded an extra point (+1), rewarding building units accessible and in proximity to access nodes. The permeability network therefore emerges from agents occupying the morphology they helped to create.

[75] Measures of density and urban form in relation to morphological metrics are contested due to their diverging interpretations (Berghauser Pont and Haupt 2004).

MIX THROUGH PROXIMITY

Footfall indicated by way-finding agents represents the key driver for evaluating and weighting the location value of buildings. The addition of a new building unit is also dependent on the score of an existing unit. When an existing unit crosses a score threshold, it checks adjacent plots for availability to build. New units can be added on three sides of an existing unit: next to, behind and opposite. Units have a minimum definition of a cubic volume with vertical faces labeled as 'side', 'front' or 'back. A new unit can only be placed adjacent when the maximum block length has not been reached. When a maximum block length has been reached, new units turn the corner of a block by being placed 'behind'. As all existing units check simultaneously, two units cannot be allocated on top of each other. Increasingly, closed blocks emerge.

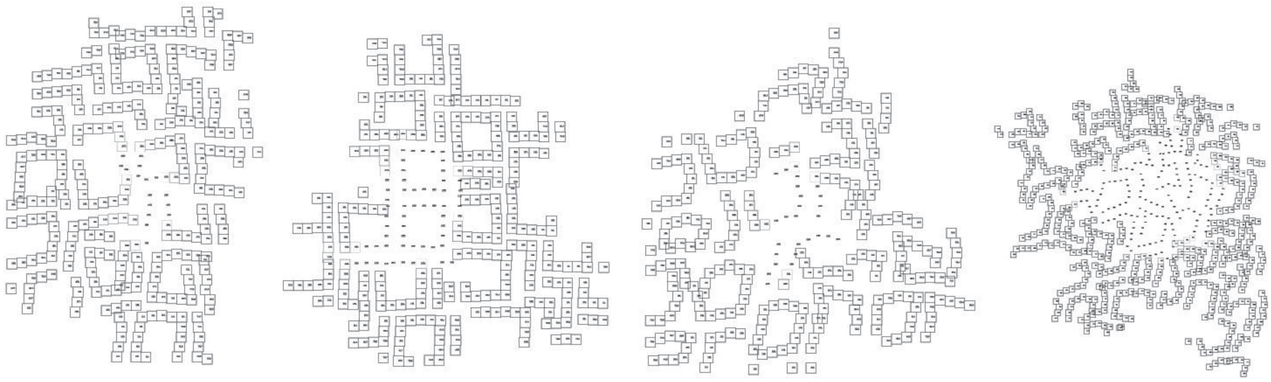


***Fig280**. SynUrb, 2004: same four states as in the previous figure, showing different urban structures emerging based on the street variables like set-backs and block lengths (numbers in cells simply enumerate their sequence of appearance); cells vary in line-colour by scale and seem to disappear in centre*

Existing units also check for available plots opposite themselves. If empty plots exist within a range, a new unit is added opposite with the facing side labeled as 'front'. All units conduct clash detection when adding new units in order to avoid building into other blocks and block streets corridors (not always successful). Each unit compiles a topological list of neighbours, including opposite facing unit across the street.

After each agent movement cycle which attributes scores to building units along footfall, a one-dimensional cellular automaton distributes scores amongst the units. Each unit sums the score from its topological neighbours, includes its own and averages this value into its temporary state. This simple transition rule is based on the popular voting rule, which is an averaging method. Unit scores therefore are equivalent to CA cell states. Neighbors of successful building units profit from topological proximity because of the score diffusion through the voting rule. The unit score informs the building scale by floor numbers (height), generating topological clusters of buildings with similar scores.

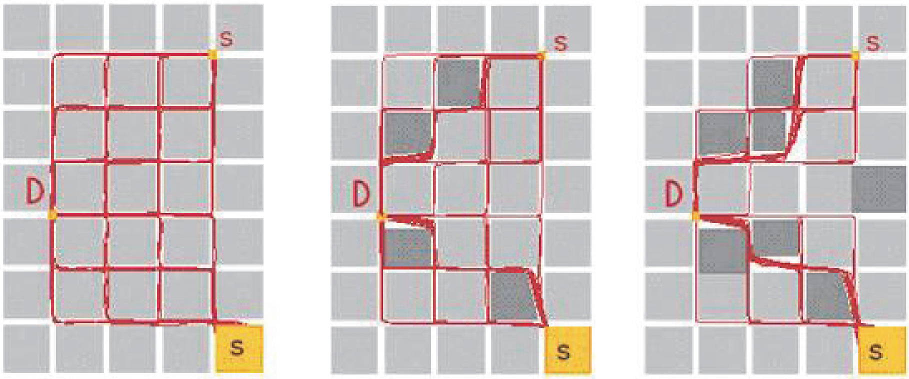


Fig281. *Christoph Hadrys, 2008: a diagram by Hadrys for an urban design at Thamesmead council, London; Hadrys runs the MA Urban Development at UEL; this diagram was used as an illustration for the heuristic of 'grid deformation' by flows, used in the collaborative SSSP project (UrbanBUZZ 2009)*

The morphology resulting from the additive process of the building units is regulated by structural variables set by the user in the GUI, mainly the maximum street width, building frontage setback and block width. Exact street widths and frontage setbacks are decided *in situ* by the building unit when adding a new topological unit, calculating tolerances from surrounding units. This is again in line with urban planning heuristics, to slightly offset building frontages and allow for larger deformations of the urban grid in the global structure, akin to a Markov process (CABE 2000).

Finally, unit scores are decaying monotonically. When unit scores fall below a threshold, the unit is erased from the morphology, making its survival dependent on its topological neighbourhood and way-finding agents. An archeological layer of demolished building units is compiled. The resulting massing reflects a history of events mediating way-finding performances of an emergent morphology and a growing topology.

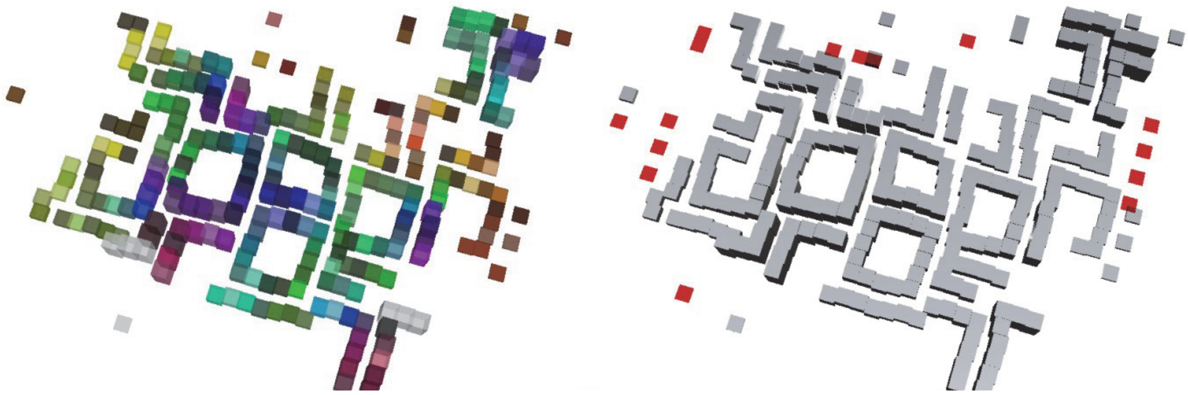


Fig282. *SynUrb, 2004: (left) a state of the system with colours indicating heights and (right) built (grey) and demolished units (red)*

DENSITY THROUGH ACCESS

The third process creates a tree graph from inserted transport nodes. Agents with low power levels searching for access nodes conduct a density check. At each step of their search they collect the density value of surrounding building units within a radius, independent of their FOV. This radius is set in the GUI, called *search square*. If no access node is available at the end of their search period and the average density level at that location is above a certain threshold, then this location is added to a queue of candidates for a potential access node. Each candidate location is checked by agents for distance to other potential access nodes representing the only pre-determined but not tautological top-down aspect. If this candidate is repeatedly confirmed by a set number of agents and it does not compete with other nodes within a set radius, a public transport access node is placed. A new node spawns a new batch of agents supporting further growth of a periphery where the access node was set.

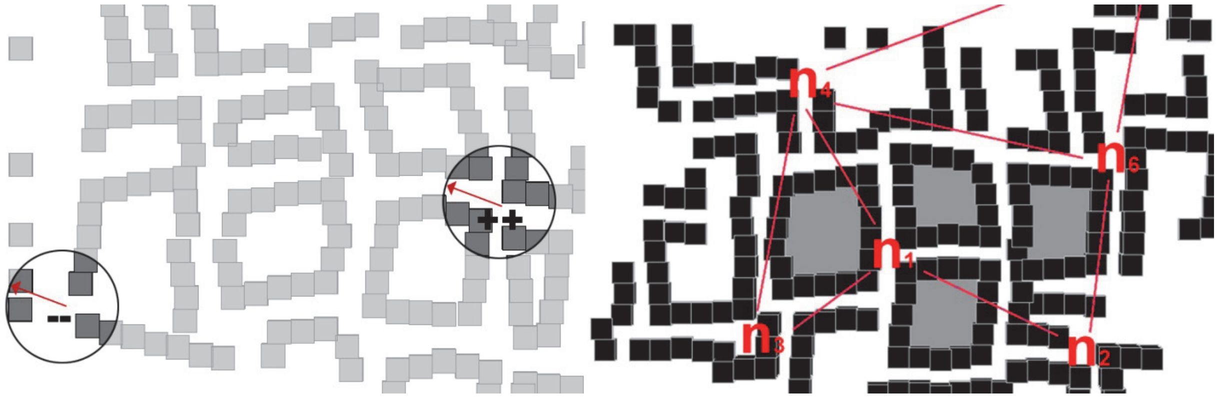


***Fig283**. SynUrb, 2004: (left) two scenarios of agents checking for the density of a location, either increasing or decreasing its chance for the addition of a public transport access point; (right) five nodes spanning across a morphology of the growing public transport node graph*

Transport access nodes grow proportionally with the expansion of the morphology and hence are extended from centre to periphery, albeit not linearly as a fixed sequence. The growth corresponds to agents being able to access newly built neighbourhoods. Inserted access nodes increase the score of building units that are directly adjacent to its geometric location, creating a feedback loop to the density cluster. An extra score is diffused via that CA into the topological neighbourhood reinforcing the value of the location. Transport nodes are given an accessibility value, which is reinforced by agents' usage. Resulting transport spanning tree graphs reflect the time-based series of their placement and the limited radius topology of expansion.

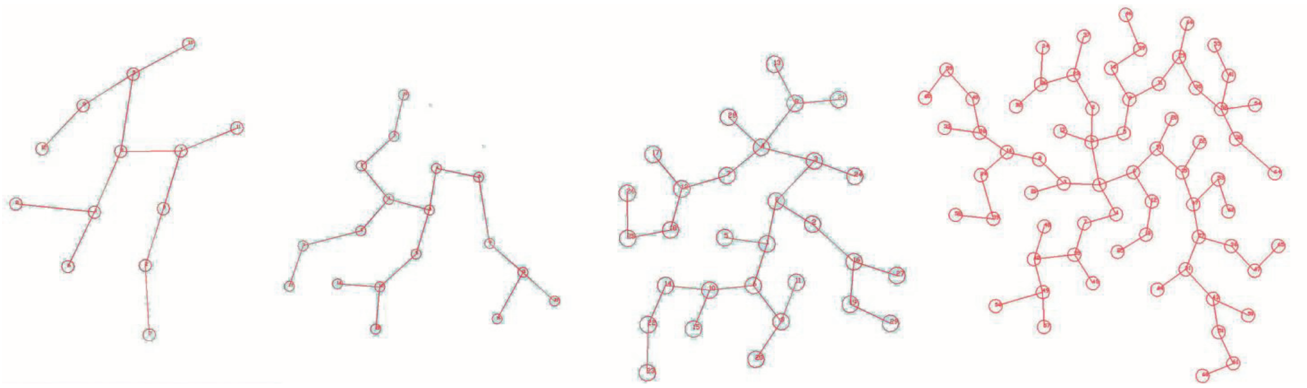


***Fig284**. SynUrb, 2004: four public transport network graphs of the same states as the agents' trails and urban structures' figures above; the connections of the graphs represent the order in which nodes were placed*

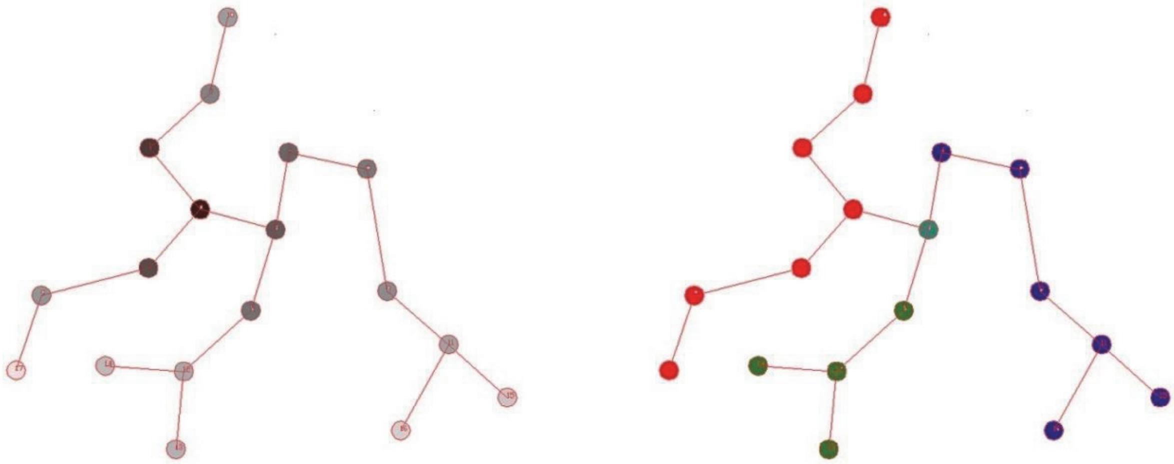


Fig285. *SynUrb, 2004: each access nodes graph grows with the urban structure; (left) the grey scales of the nodes indicate the chronology of insertion and (right) three sub-branches of the graph into distinct sub-areas of the morphology*

DOMAINS

A complex set of feedback loops generate the urban morphology diagram, based on a series of associations between aspects of urban form and dynamics. While the morphology may not be large scale or contain hierarchies of spatial differentiation, various aspects of urban form are emerging simultaneously within a self-regulating system. An equilibrium of occupation, configuration and morphology is perpetually maintained, generating and eroding its own structural elements as in Maturana's autopoietic concept of *structural coupling* between systems (Maturana 1978).

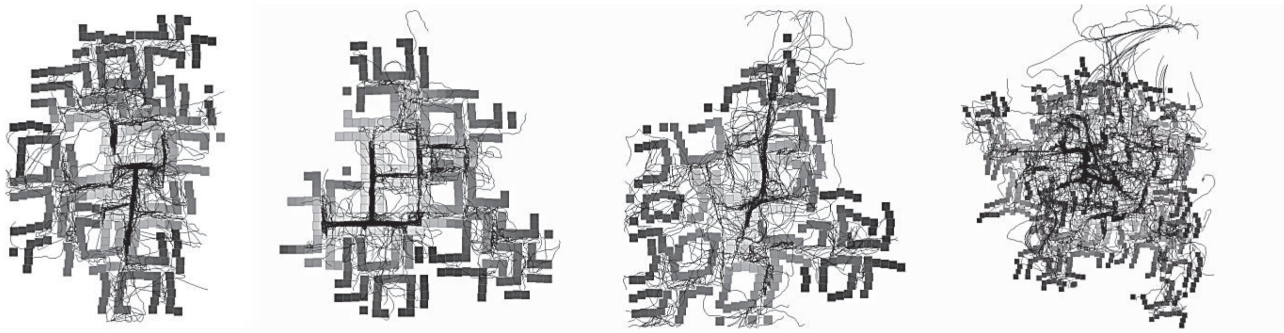


Fig286. *SynUrb, 2004: the four previous states of agent trails overlaid with the grown street network, showing clearly how movement patterns and morphology reinforced each other*

From this coupling some domains are created that go beyond hard-coded associations. The code does not explicitly align movement with streets that emerge over time. Agents use a perceptual property and social interaction to navigate while not having alignment to streets encoded. Neither do building units 'know' of agents. A consensual domain is created where some agents' actions unintentionally provide cognitive affordances to other agents affecting their behaviour. The consensual domain is perceived by the observer as an integrated single expression of the field. Its autonomy from the underlying processes evokes the principle of *soma-tectonic* communication or simply *stigmergy*. *Stigmergy* was named by the biologist Jean-Pierre Grassé to describe the social collaboration of insects like termites that communicate indirectly via their interactions with the environment (Theraulaz and Bonabeau 1999). Individuals read the state of the environment manipulated by previous individuals and respond instinctively. The field as environmental construction therefore is not directed but self-organizing. Equivalently in this SynUrb system, the parts of different processes collaborate indirectly by constructing a shared field of affordances.

Another unprogrammed domain emerges temporarily around access nodes. Despite the hard-coded agent behaviour for placement of access nodes, network nodes are dependent on successful continuous accessibility, which in turn is subject to a growing morphology. Temporary attractors are formed by some access nodes that are more accessible than others at central places between newly grown areas. Those local nodes and street corridors leading up to them dominate the global morphology temporarily until new attractors emerge. This competitive dynamic reflects Haken's (2004) *enslavement principle* and thus the system can be said to display synergetic properties.

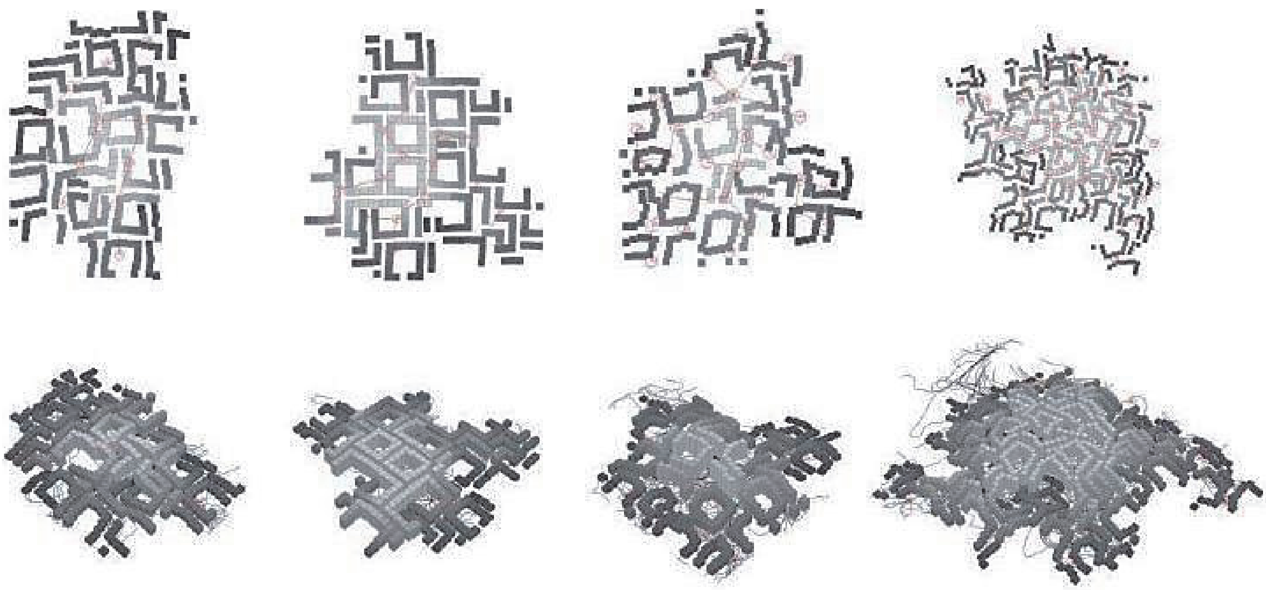


***Fig287**. SynUrb, 2004: the four states as plan and perspective mass, showing the grown center-surround structure*

As stated above, the resulting morphological states are not of large scale and do not differentiate into complex sub-structures. Equally, streets do not emerge as perfect road networks or robustly permeable links. However, compared to Hillier's and Coates' alpha syntax models or the top-down parametric urban planning models mentioned above, the permeability network and morphology manage to display similar complexities, despite being autonomic. An additional dimension of complexity added in SynUrb represented the intentional omission of a development grid, which usually enforces morphological alignment. Without a grid, spatial propagation of differences is more nuanced and enables more rigorous conjecture testing about spatial constraints at this scale. When Paul Coates first saw the model he recalled Bill Hillier suggesting that when some global superstructure could emerge regulating local configurations then a truly autonomic system would be achieved. Coates suggested in a conversation in 2005 that this model was a first proof-of-concept that Hillier's suggestion was viable.

8.2 REFLECTIVE SEARCH | PARTS-FIELD-OBSERVER MEDIATION

"He reflects on the phenomenon before him, and on the prior understandings which have been implicit in his behaviour. He carries out an experiment which serves to generate both a new understanding of the phenomenon and a change in the situation." (Schön 1983, p68)

SynUrb provided an example of an autonomic algorithmic design system without the observer in the loop. How to include the observer into an autonomic design system with human-centric spatial properties? Chapters 6.2-3 and 7.2 provided examples of the observer as situated designer and user participating in an algorithmic model, albeit either as global agent or as local mediator being hierarchically superior to the generative and analytical algorithms. The mediation consisted usually of a linear correspondence between some action by the observer and some reaction by the algorithmic model. SynUrb already overcame this linear action-reaction sequence by opening multiple algorithmic models onto each other into an autonomous feedback system yet excluding the observer. To subsume the observer into an algorithmic design system the observer should lend some cognitive agency while also mediating as an equivalent component within the process. Knowledge from observer and meta-heuristics must be synthesized into one, creating an epistemically autonomic system.

Donald Schön (1983) found that architects as designers function as observers of phenomenal states during the generative process (*reflection-in-action*). The reflection on their actions introduces their learned yet implicit knowledge about spatial phenomena and of *being in space* like an internal simulation. Iteratively the observer as designer converges onto a *generative metaphor*, which provides him with a design intention, by which to evaluate his actions further (Schön 1983, p185). Here Schön intended a workflow and its knowledge-generating property rather than the structure of a system.

Bill Hillier (1996) equivalently researched the epistemology of a syntactical workflow in Space is the Machine. Using the *barring process* analogy for his concept of *emergence-convergence*, Hillier proposes that the *pre-structures* of a designer as observer and user of the world inform his design strategies that inevitably will lead towards some spatial configuration encapsulating not so much a building typology but a *generic function* (see 2.7.3). Pre-structures represent the designer's cognitive and behavioural heuristics used to develop design strategies. Those in turn can be encoded into representational mechanisms like algorithms and, if more complex than a simple *barring process*, eventually into a system. The *barring process* successfully illustrated how each move (or *action* in Schön's terminology) produces a global state that the observer-designer evaluates (or *reflects* on) by applying his cognitive experience. The evaluation leads to some perceived concept of spatial typology based on the observer's experience of occupation and provides the constraints on the next actions to approximate that typology. As a typology of occupation emerges, the number of possible actions converges. Hillier called this the *local-global law*, governing configurational performances that in turn encode typologies of occupation, or *generic functions*.

This section discusses a project that attempts to develop a reflective workflow based on pre-structures of the observer's cognitive and behavioural experiences. Using Hillier and Leaman's terminology (1974), the development of a *manifold* was aimed for that implements the *emergence-convergence* concept. The design brief specifically excluded the setting of a schema or typology but to converge towards an occupational typology.

8.2.1 Implicit Space

During a guest-professorship at the Technical University Munich from 2011-12, the author managed an institute with a design studio, lecture series and two technical modules[76]. The unique opportunity arose to concentrate a design studio, theory, and technical modules on a single brief. A programme was set out to introduce students to algorithmic thinking, spatial theory and learn computer programming in order to design a bespoke design system within one semester, i.e. 12 weeks. The programme, called Implicit Space, foresaw three phases of development:

A. Reveal — correlations of spatial dynamics and occupant's experience
B. Encode — behavioural rules and cognitive associations into spatial patterns
C. Compose — a system for spatial organizations based on A + B

The title Implicit Space was meant to make students aware that our observations and use of buildings encode spatial dynamics that can be abstracted into associative rules. Design heuristics underlying standardized design procedures contain implicit spatial expressions, which can be challenged to produce any expression of human-centric spatial configuration based on subjective experiences (Derix 2012).

Fig288. *Hochschule fuer Film und Fernsehen, Boehm architects, Munich, 2012: interior shots of the atrium and the 'stairway to heaven'*

In phase A, students had to collect subjective data on the occupation of a building and create maps of correlations between space and their personal perception. The Munich School for TV and Film (HFF) with a range of spatial qualities served as case study. The maps sought to reveal spatial features, dynamics and intensities that make up a phenomenon or experience. The resulting maps had to be abstracted to serve as generative diagrams.

[76] https://emtech.wiki.tum.de

In phase B, rules of the diagrammatic mappings were to be extracted and encoded into computer syntax, using the Java language of the Processing environment. Some examples were provided during workshops such as agent-based behaviours or cellular automata.

Phase C was conducted in groups and subdivided into two stages: 1) developing mixed-media components of cognitive phenomena into a catalogue and their generative rules and 2) synthesizing them into a design system to eventually create a design instance. Mixed media refers to different representations of aspects of spatial composition because limitations of the students' computer programming skills were expected. But primarily, the constraint to construct a design system mixing media aimed to allow students to explore spatial qualities through appropriate representations. Not all phenomena can be investigated digitally and a link between haptic properties of physical models and dynamic behavioural properties of computational models was desired. It was demanded that experiential qualities had to be quantified via one of two representations that are combined through diagrammatic processes into a workflow. A three component workflow results that correlates: a modular catalogue of cognitive scale models, algorithmic rule-based evaluation and an analogue synthesis. Because the algorithmic evaluation and the cognitive scale models had to be generalized before integration, resulting systems embodied a meta-heuristic synthesis.

FLOATING ROOM

Six groups developed a design system with varying degrees of success for integrating the mixed heuristics. The project discussed here was computationally not the most advanced but the synthetic workflow was sophisticated.

The title Floating Room indicated that there should be a distinction between the standardized graphical representation of a layout plan and the perceived boundaries in space. A 'room' does not necessarily correlate its wall boundaries with its cognitive and behavioural domain. The project set out to develop a design system to facilitate the planning of residential layouts relating spatial boundaries to effects of cognitive properties such as visual exposure.

A - Reveal

Students arrived at this brief by mapping their sense of visual exposure to spatial conditions at the HFF. The scope derived from the compatibility of their observations that the sense of privacy correlated to the configuration of space facilitating degrees of exposure.

Although the four students had used a similar phenomenon each mapping revealed different aspects: Katariina Knuuti chose the impact of visual connectivity on her orientation in space; Jana Baeumker tried to correlate visual connections with spatial depth; Juan Carlos Venegas del Valle explored levels of claustrophobia in relation to the area of view and rhythms of external visual connections and Anna Wojciezek looked at levels of perceived exposure in relation to spatial sections. The maps were consistent but subjective, providing four valid interpretations of visual conditions through the students' personal *pre-structures*.

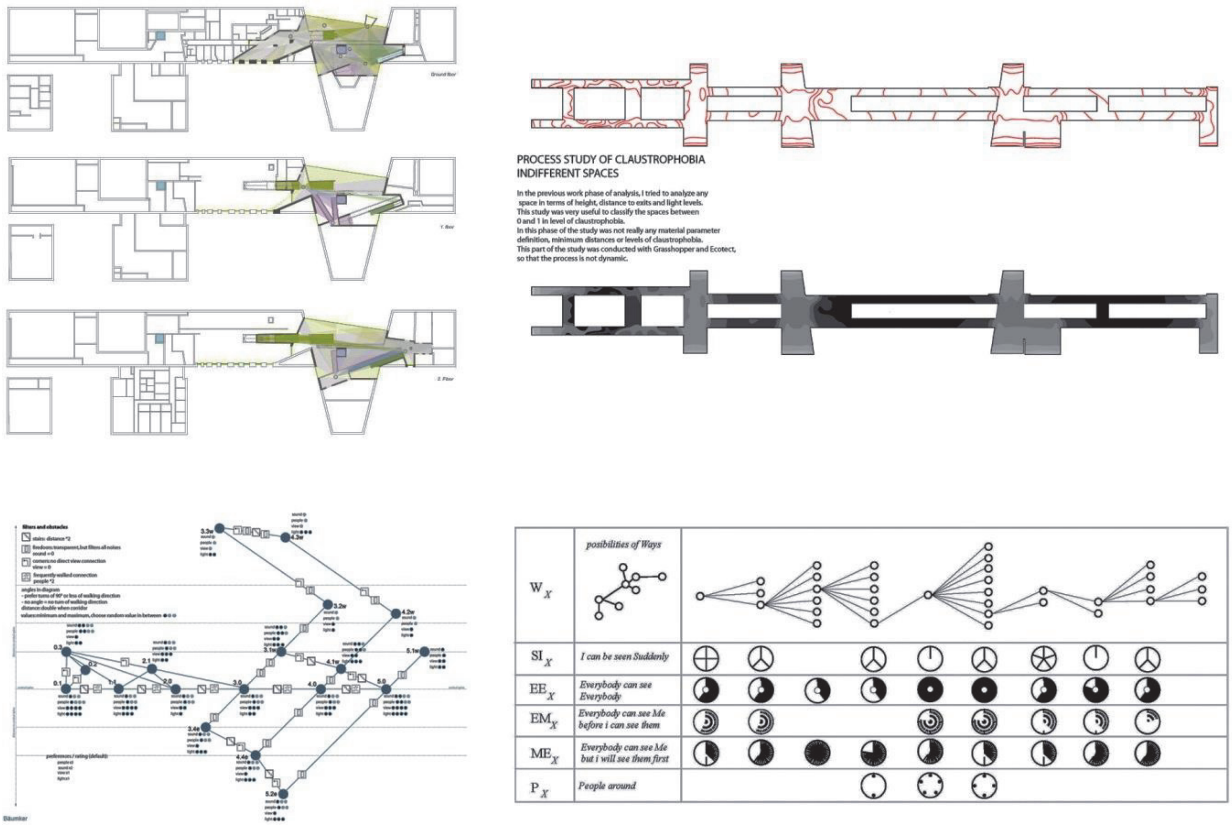


***Fig289**. Floating Room, 2012: four observational mappings by (top left) Katariina Knuuti analysing consecutive visual access; (top right) Juan Carlos Venegas Del Valle mapping levels of perceived claustrophobia; (bottom left) Jana Bäumker mapping movement filters and (bottom right) Manuel Gmoll mapping critical exposure locations*

B – Encode

Some maps were successfully generalized into generative diagrams encoding decision rules that allowed a regeneration of the original experience into new contexts. Student groups were to find a real-life analogy to a spatial environment that encapsulated the properties of their observational maps and generative diagrams. A *generic function* was thus anticipated as correlation to the cognitive and behavioural dynamics revealed. The Floating Room group chose several small scale buildings, particularly residential spaces such as homes.

***Fig290**. Floating Room, 2012: programme analogies deriving from the observational mappings and behavioural codes, including open plan and discrete partition examples such as Mies van der Rohe's Barcelona pavilion (1929) or the Four Corners house by Avanto architects, 2011*

Knuuti and del Valle encoded their observations through versions of the discretized visibility mesh. Knuuti's employed a biased isovist that mapped only viewpoints different from the viewer to show how much a space is visually merging with others, diffusing clear geometrical boundaries. Del Valle applied an isovist weighted towards selected edges of the perimeter to show how proximity to closed or open edges influences the feeling of claustrophobia. Gmoll (who wasn't part of the group but worked with similar principles and informed the group's work) elaborated a protagonist-exposure isovist, where the protagonist would be mapped by the amount of positions he is exposed to.

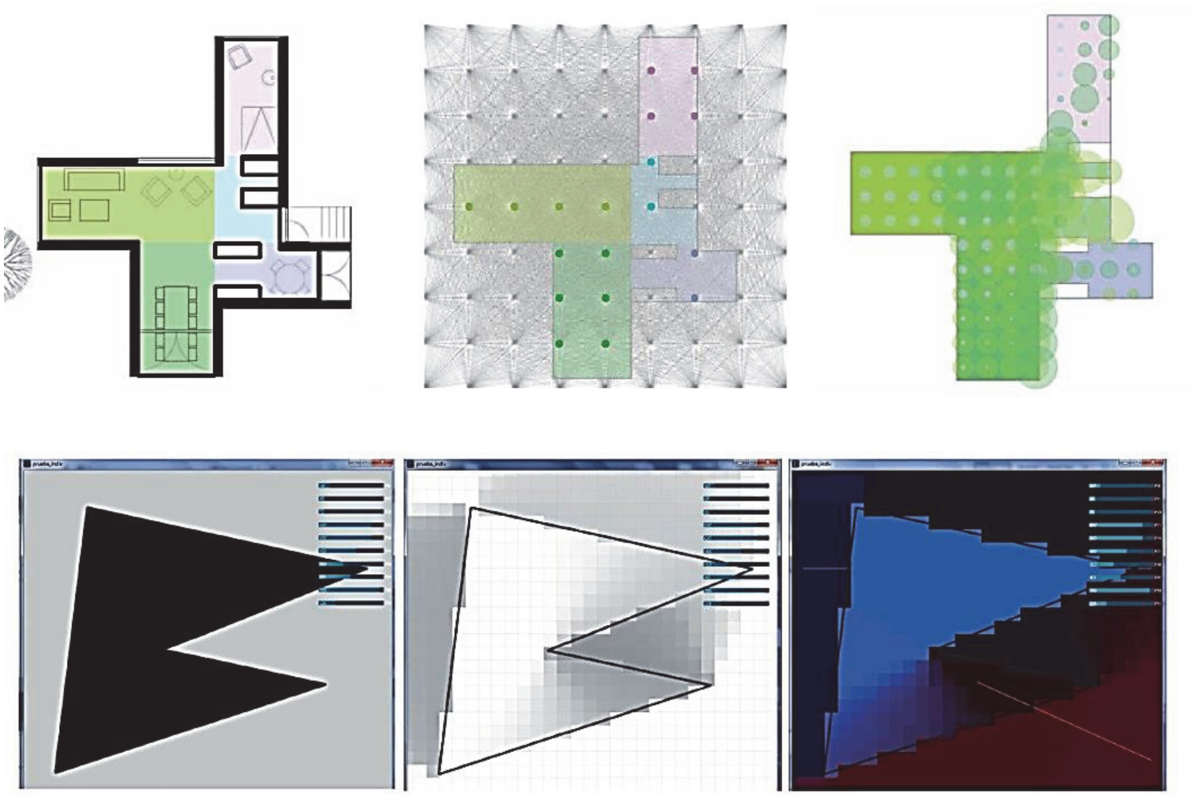


Fig291. *Floating Room, 2012: rule-based abstractions of mapped observations by Katariina Knuuti (top) Juan Carlos Venegas Del Valle (bottom); Knuuti merged the discretized visibility isovist mesh with room exposure; Del Valle weighted the discretized isovist with proximities to edges*

The Java coded prototypes served to identify behaviours and cognitive qualities that were to specify the design brief. The simplest structures for the laws of aggregation were extracted as Hillier and Leaman suggested. Those structures as correlation between space and perception of occupation provided the syntactic strategy to be encoded into the representational system to drive design decisions. Cultural differences played an important role, as Knuuti - a Finn - or Del Valle - an Argentine - had very different perceptions of privacy.

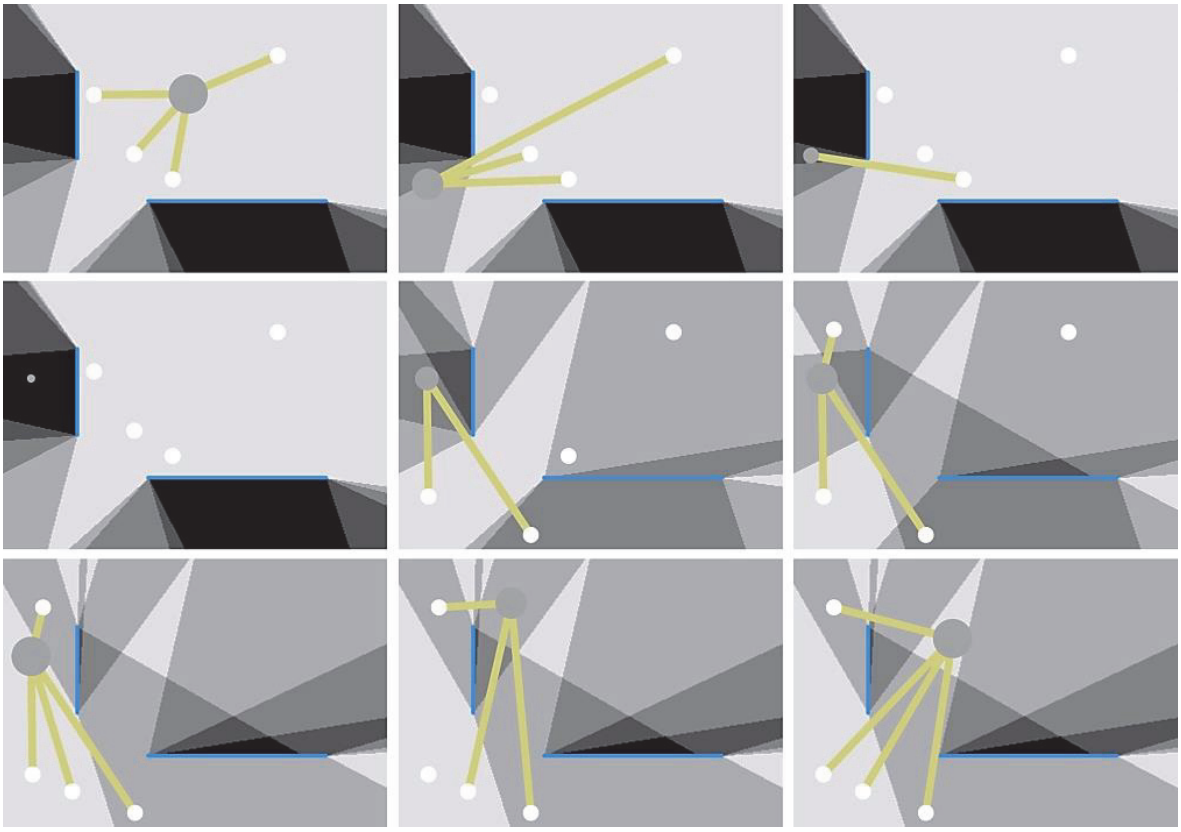


***Fig292**. Floating Room, 2012: Manuel Gmoll's mapping and coding informed Floating Room with is 'exposure' patterns; the series shows a protagonist position and its exposure to four other points; the grey scales map the amount of visible space by those four points*

C – Compose

Eventually, a brief was formed based on principles of visual exposure, private activities and intervisibility or inversely, the lack of it. The students summarized the design KPIs to be: "*protected vs open – privacy behind corners – limited visual connections forming space*". The brief identified a single family apartment with the hypothetical client constituting a family with two children. The site was a residential block on the outskirts of Helsinki, Finland, with two external facades oriented east-west (access from east). A connectivity diagram and 'exposure matrix' was drawn up from the client profile, providing information about privacy for each space type. All spatial partitions were attributed a *floatiness* value consistent of four visual properties:

- Visual connectivity (to other positions on the grid)
- Visual overlap of functional partitions (rooms)
- External views (length of visible perimeter and orientation)
- Area of seen space

The *floatiness* value resulted from weighting the four visual properties (brackets indicate semantic performance description)

- 50% visual connectivity and area (privacy & continuity)
- 30% visual overlap of functions (non-monotony)
- 20% external views

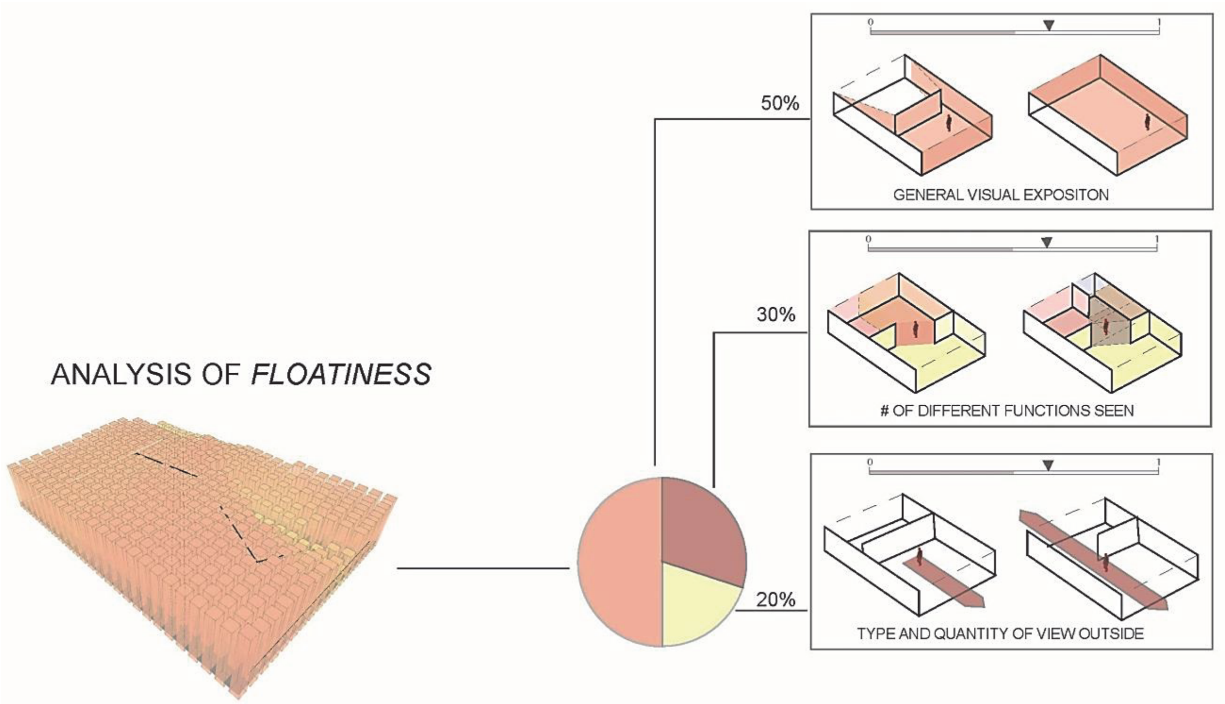


Fig293. *Floating Room, 2012: the composition of the Floatiness value as an evaluation of a configuration; the value is weighted by four visibility measures into three variables*

Strategically, it was decided to follow a variation of Hillier's *barring process*, by iteratively placing partitions to represent wall segments. While a room schedule was identified for the clients, the allocation of rooms was to emerge from the spatial conditions generated by each wall placement, mirroring Hillier's concept of *emergence-convergence*. Finally, all components were organized into a generalized workflow to produce spatial configurations that mediated experiences of the initial spatial observations.

The encoding phase had given rise to mostly planar representation of spatial conditions. The algorithmic representation provided a syntactical and thus operational evaluation method that could be elaborated into one component of the workflow. But it only represented one dimension of the *simplest structures*. A three-dimensional model of the two-dimensional planar configuration was elaborated via physical scale models. Scale models were not simply meant to check the extrusion of the plan but calibrate potential design moves of planar partitions in order to construct a catalogue of permissible moves. What appeared possibly as a meaningful result from the computational exposure analysis might show up to be an irrational condition when inspected volumetrically.

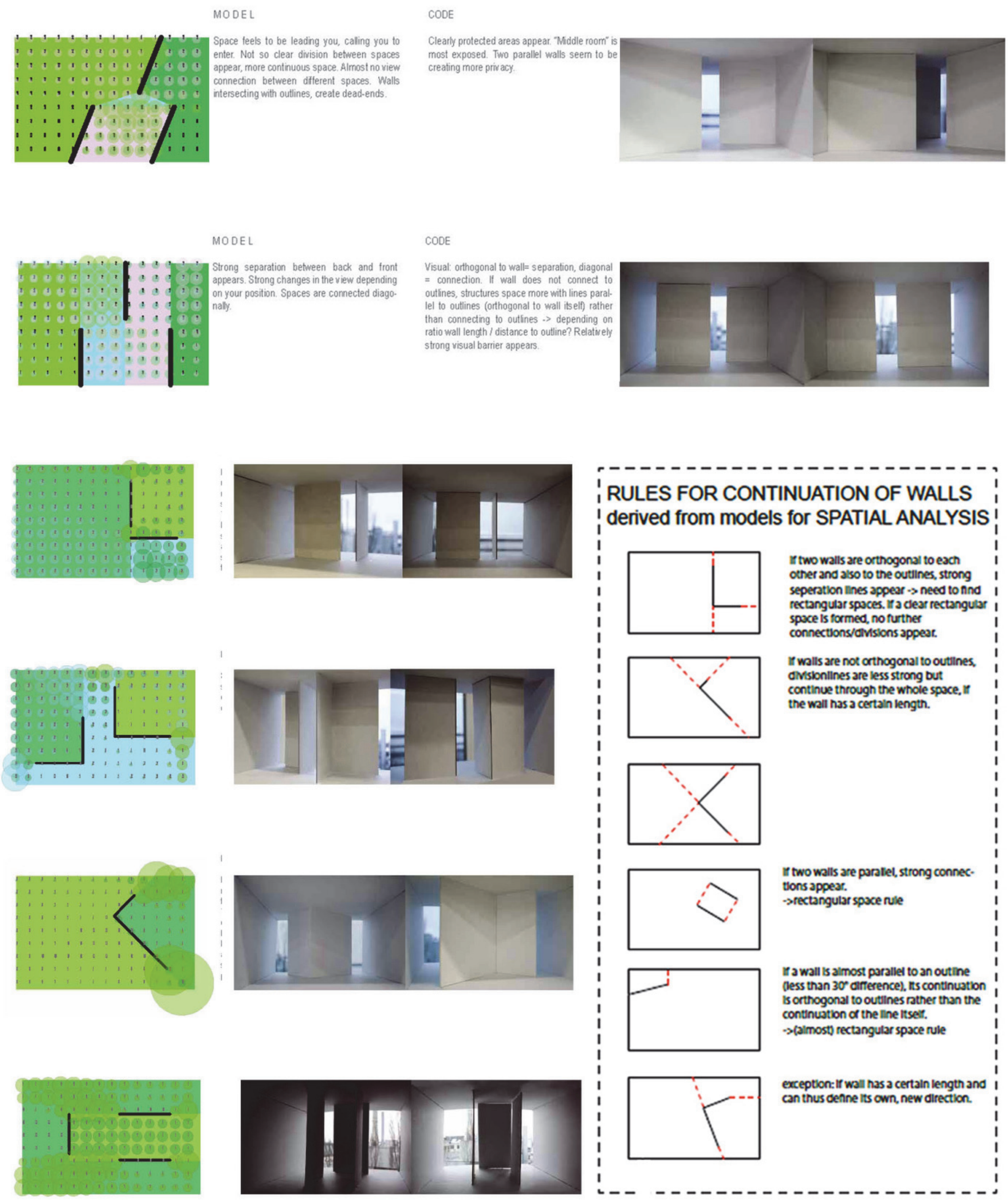


***Fig294**. Floating Room, 2012: (top) two instances of the catalogue from the calibration phase between computational algorithm and physical scale models. The text in the middle identifies mutual qualities and recommendations. And (bottom left) four instances of comparison between algorithmic analysis and physical models, which led to (bottom right) the specification of permissible wall placement moves of the design system.*

Configurations were evaluated for conditions subject to symmetry of placement, angles between partitions, length ratios, proximity to perimeter (mainly natural light and view) and continuity. Continuity and privacy represented the most dominant drivers as they would best inform permissible moves, benchmarked against the client profile KPIs. A physical catalogue of conditions was compiled representing favourable states as relations between partitions. The physical catalogue constrained the partitioning moves and informed emerging configurations, analysed via the visual exposure algorithm. A workflow crystallized where the observer would be subsumed into a set of generalized correlational heuristics.

This meta-heuristic workflow can be summarized through the following steps

1) Consult *floatiness* values of area schedule set in brief
2) Place partition based on permissible patterns of spatial properties
3) Evaluate continuity from catalogue by inspection and define areas
4) Evaluate four states of visual conditions
5) Refine areas by comparing emerging conditions to KPIs from brief by:
    a. Adapting resulting areas from 3) and repeat 4)
        OR
    b. Placing new partition = loop to 2)
6) Generate *floatiness* value for whole configuration and loop to 1)

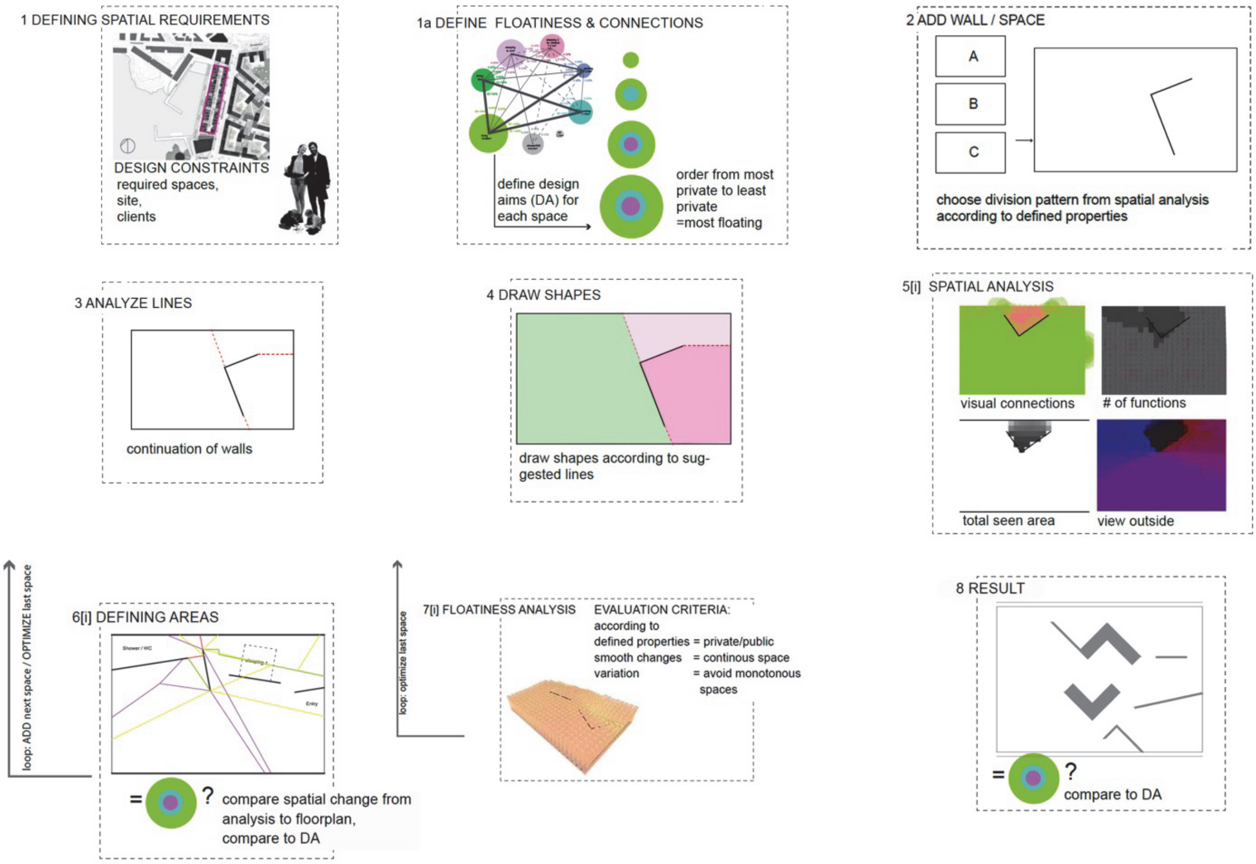


Fig295. *Floating Room, 2012: 6(7) workflow stages as described above with 1) not part of iteration and 8) coinciding with 1)*

The system was implemented and three instances of apartment layouts produced. Decisions and configurational steps of each instance were monitored and archived by exporting all partition placement and evaluation cycles.

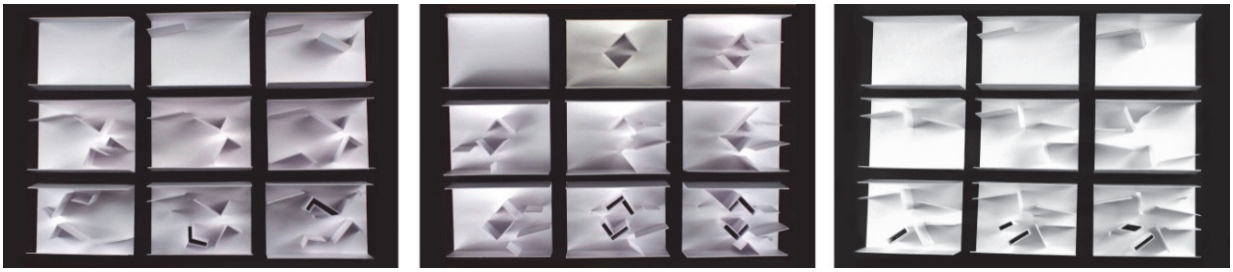


Fig296. *Floating Room, 2012: three instances produced from the design system, showing nine iterative steps with the first and third instance's first partition placement in the same location but diverging continuation; the three instances respond to three selected residential units in the case study building block with varying external conditions (see final block model photo below)*

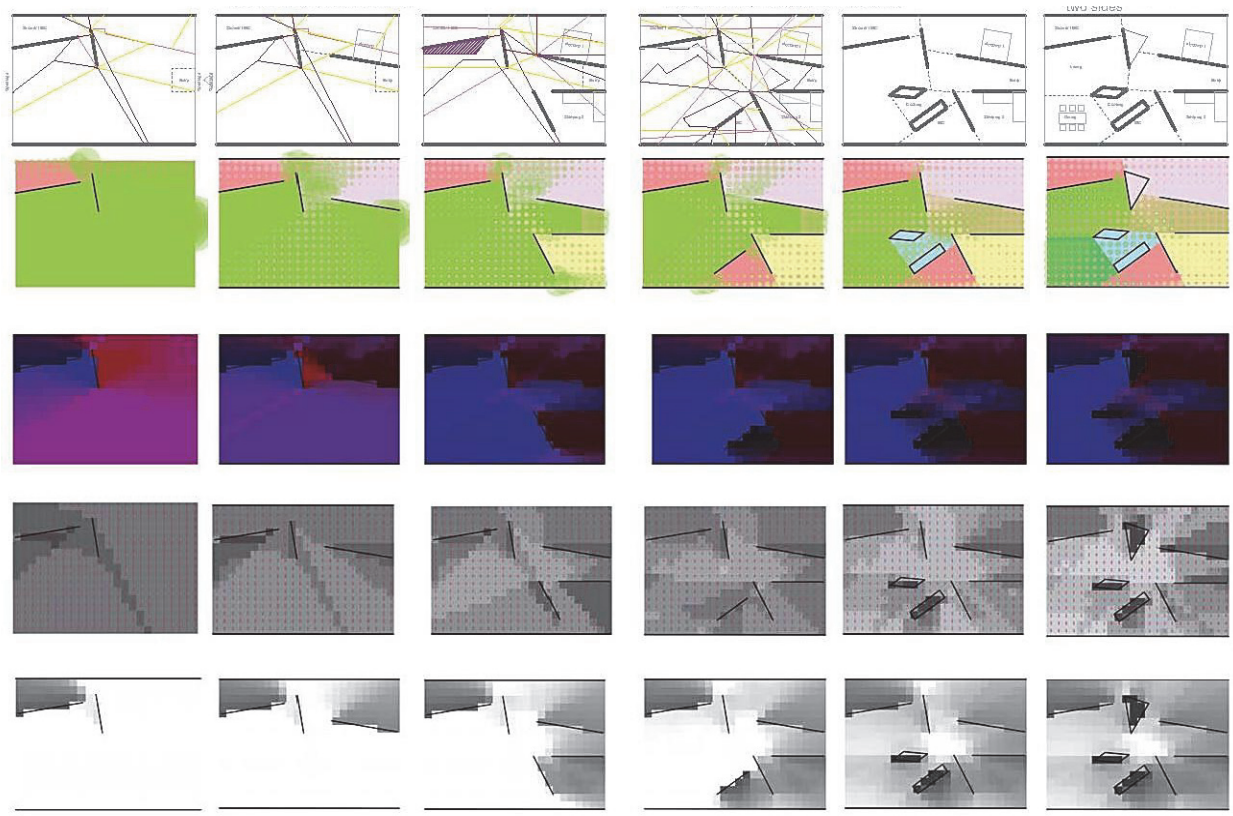


Fig297. *Floating Room, 2012: the development process of instance 3 (see above), showing the placement of wall paritions, evaluated for the four visibility criteria and area definitions (top)*

SYNTONIC SYSTEM

"Jean Piaget's work on genetic epistemology teaches us that from the first days of life a child is engaged in an enterprise of extracting mathematical knowledge from the intersection of body and environment. The point is that, whether we intend it or not, the teaching of mathematics, as it is traditionally done in our schools, is a process by which we ask the child to forget the natural experience of mathematics in order to learn a new set of rules." (Seymour Papert 1980, p118)

Floating Room was only one of six design systems that encoded *pre-structures* from subjective syntactical diagrams into a generalized meta-heuristic workflow. The results emphasized that experiences of space share behavioural and cognitive code that although generalized and apparently objective, can utilize personal metrics and syntax. The system produces empathic configurations that re-produce the observations of the observer as occupant mediated by heuristics of the observer as designer. Distinctions between the components of the system are eroded. The observer is not embodying a global role outside the system but his creative routine is placing lines and comparing results. Essentially, anybody else can execute the system re-generating the experiences of the original observer.

Seymour Papert employed the concept of *syntonicity* from psychology to describe an emotional responsiveness to the environment (Papert 1980). Papert used syntonicity as a model of representing the situated observer in the world and distinguished between *body-* and *ego-syntonic*. Body-syntonic learning describes empirical knowledge through physical interaction with the environment, which is abstracted in our minds through ego-syntonic mental states. Papert believed that algorithmic rather than mathematical representation helps the observer to create a new epistemology of the world. Algorithmic meta-heuristics are thus a vehicle for empathic planning, correlating spatial with social behaviours, echoing Hillier's *inverse law* (Hillier et al. 1976, p179; Hillier 2014)*.* The Implicit Space semester demonstrated how to extract rules of natural experience and encode them into an algorithmic design system (Derix and Izaki 2014).

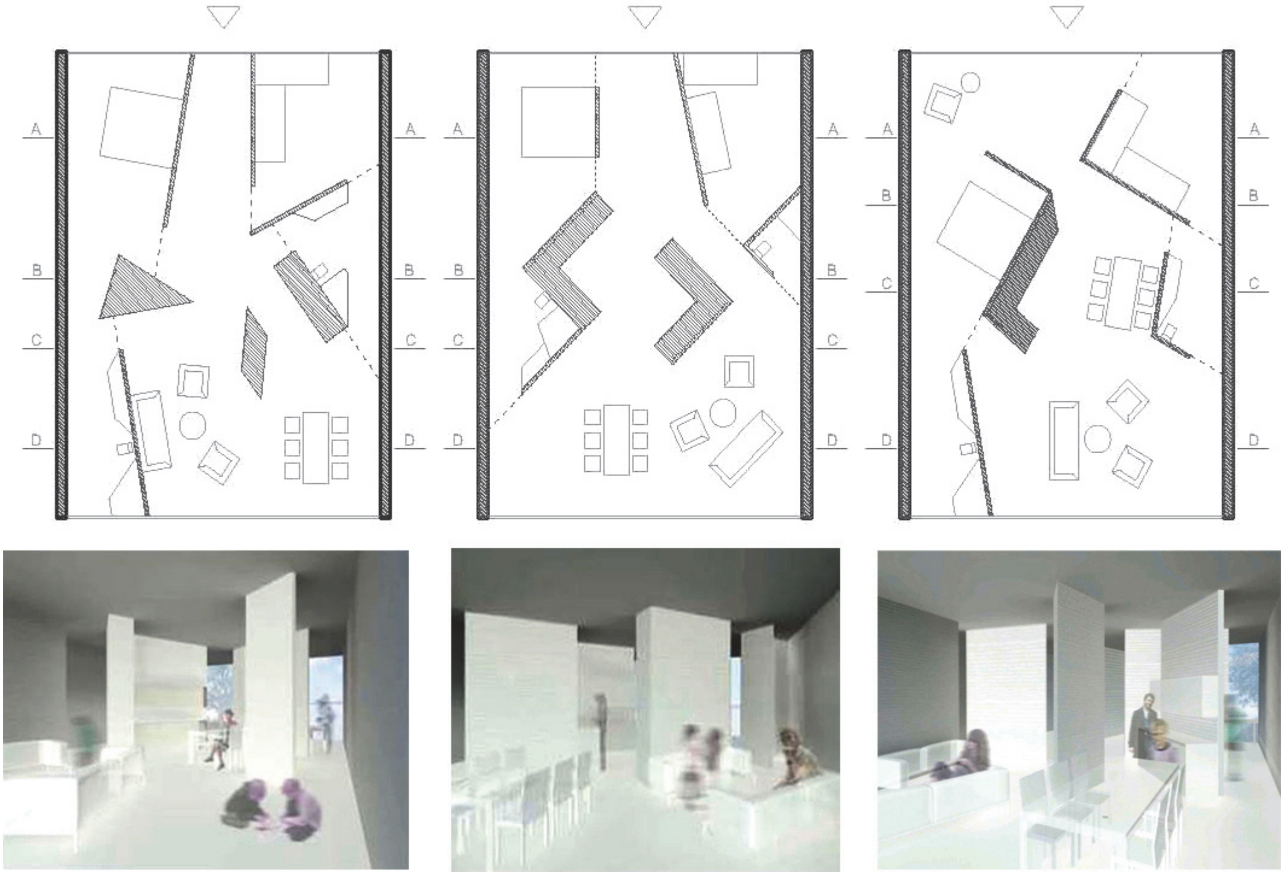


Fig298. *Floating Room, 2012: instances 1-3 drawn as final layouts with furniture icons indicating room types (top); the top arrows indicate the apartment entrance locations; and (bottom) renderings of interior views of apartments*

8.3 OPEN SEARCH | SPACE-OBSERVER FRAMEWORK

"Finally, a complete examination of the implications of field conditions in architecture would necessarily reflect the complex and dynamic behaviours of architecture's users, and speculate on new methodologies to model program and space." (Allen 1997, p.26)

Two general categories of design models associating generative algorithms with human-centric affordances have been discussed from chapter 5-8: the remote and situated observer (the third category of chapter seven *Learning Observer* being an extension). Remote Observer models separated the observer in two ways from the generative components: the observer as designer guided the generative process globally; and human-centric affordances were used by the observer as explicit evaluation functions. The design model is reliant on the meta-heuristic and thus epistemic structure of the generative algorithm.

The Situated Observer models require the observer as designer to mediate human-centric affordances and generative processes. The observer as mediator is either reactive to the human-centric affordance evaluation or proactively mediates generative and analytical components. In both cases, components that evaluate human-centric affordances provide the dominant epistemic structure for the design model. So, essentially four types of model structures exist:

- Generative meta-heuristic led (remote observer)
- Observer-designer led (remote observer)
- Analytical meta-heuristic led (situated observer)
- Observer-mediation led (situated observer)

All but the last case study model (Floating Room) represented stand-alone applications (or *macro* or *plug-in*, depending on programming environment). The relationship of the observer was hardcoded, attributing specific *a priori* steps or roles in the design process. Observer roles for specific agencies usually relate to a concise product description like the VITA shelving system, or represented exactly one component of a larger system such as FuCon II or SynUrb. Hardcoded roles of each component reflect a trade-offs between (meta-) heuristics of the generative and analytical algorithms and the observer, encoding the dominance of one of the heuristics. Consequently, the epistemology of one component implicitly controls KPIs (even when the remote observer sets explicit targets), because schemata and performances driving the design model are affected by the structure of the dominant meta-heuristic. In that sense, all case-study models resemble academic models which generally are structured into a single monolithic system with a specific purpose, not appropriate for *designing* flexibly (Liggett 2000).

Floating Room on the other hand, approximated an open system where no individual process component dominated the knowledge-generating capacity of the synergetic design system. Behavioural and cognitive affordances inversely correlated to design heuristics, drawing closer to the intended paradigm set by the blend of Hillier's and the New Epistemologists' concept of autonomic *designing* systems. In other words, when two or more open systems would organize into a consensual field rather than being hardcoded into one model, their ability to find correlations improves.

Floating Room managed to keep the design workflow as open as possible while loosely relating associated spatial and occupants' performances. Generally, when design spaces becomes more complex with increasing numbers of constraints, behaviours and functionality, the typical reaction is to hardcode all associations into pre-determined ontologies, reverting back to Alexander's hierarchical systems theory (1964) as implemented by KbD or contemporary models of design that are specific to a typology like CityEngine. This approach loses all the benefits discussed in chapter two and undermines consensual domains or creative design by fixing deliverables and generally constraining the observers' heuristics both as designer and as simulated user. The second objective of the dissertation asks how to deal with an increased design space complexity and the inverse correlation between design and human-centric performances that require an open system proposed by many like Rittel and Webber in 1973.

8.3.1 System of Systems | Meta-System

The standard approach by industry is either to build expert systems like academia that resolve exactly one specific design problem[77] or disaggregate all design aspects into manually driven compositions, as proposed by Paul Richens (1994) at the Martin Centre at Cambridge University.[78] But neither approach has arrived at a synthetic system for a USOM. A middle ground appears the most promising route but does not provide a meaningful short-term target for academia or industry. The middle ground is approximated by *Floating Room* and the below discussed selection of instances from the Open Framework for Spatial Simulation (OFSS) like FuCon*,* RIBS or Expo 17 Village. The workflow is characterized by heterogeneous components loosely organized into a system that implements observer and algorithmic heuristics synergetically, converging towards configurational states of space. The workflow is neither monolithically integrated through hierarchies of biased components, nor is it completely disjointed by conceptual domains. The development of such a workflow for a live project setting in practice would be too slow and inflexible. Project-based workflows are exposed to rapid adaptation to changes in programme or concept, requiring an agile re-organization with lighter components that can be culled, adjusted, extended or replaced. Ideally, functional elements such as I/O formats must be interchangeable without fixing contained ontology.[79] Equally, the role of the designer-observer as allocated in Euston Crossing or Floating Room must be more flexible to mediate between several heuristic components, avoiding a biased role within the system. The picture of a single system gives way to a more generic agile

[77] Constraint-solvers such as discussed in Axel Kilian's PhD thesis for example are mostly used for solving geometric and structural design issues but are akin to Christopher Alexander's first-order cybernetics approach for spatial design (Kilian 2006)

[78] Richens' approach (1994) constituted a constructive response at the time of KbD's over-elaborate knowledge automation and could be regarded as a paradigmatic forerunner that has not been realized.

[79] The call in industry is usually for an aligned ontology of I/O formats closing systems into exclusive typologies.

system of systems where multiple consensual domains could emerge, based on heterogeneous ontologies/ schemata.

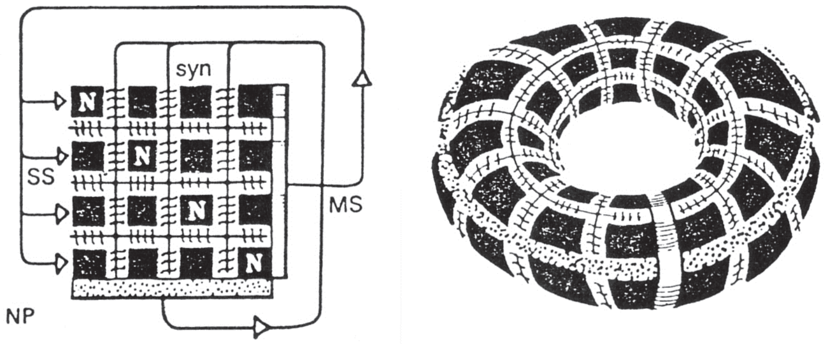


***Fig299**. Second-order Cybernetic System, von Foerster, 1984: diagram of synaptic loops between types of neurons – actuators and sensory (left), forming cyclic feedback-loops that can be expressed as a toroidal surface of computation (right); the system is closed structurally but operationally open to form new instances (schemata)*

The phrase 'System of Systems' is a reference to von Foerster's second-order cybernetics (1984) and Maturana's autopoiesis (1970). Von Foerster defined the epistemic domain of computers as *meta-programs* while Maturana (1997) called the interaction system between humans and computers *meta-design* . Both implied an autonomous system of generic programs to interact with the world through instantiations of specific programs, generating emergent consensual domains.[80] If *specific programs* a replaced with *heuristics* and *generic programs* with *meta-heuristics*, then the notion of a framework of meta-heuristics arises. Implementing this conceptual analogy, a computational design framework takes shape that should allow for fast synthesis of meta-heuristics (general) into a situated heuristic system (context specific). The declared division into *routine* and *creative* design by KbD would have provided a meaningful distinction if those creative heuristics associating routine meta-heuristics had not been automated (Akin, 1998). The middle ground for a computational *designing* system should thus reflect (a) Richens' flexible open network approach with (b) KbD's routine design meta-heuristics as components using (c) Steadman and colleagues' representations to build (d) autonomic systems as developed by the New Epistemologists and (e) correlating spatial configurations to human-centric heuristics as trialled by the Hillier and colleagues. This section introduces the Open Framework for Spatial Simulation (OFSS), which has been in development for over a decade into a robust *system of systems* or *meta-system*.

8.3.2 From Integrated Simulation to Open Planning System

The realization that a computational planning system should not integrate and hardcode its components arrived in 2008 during the development of SSSP (see 5.1, 6.1, 6.2 and 7.1). SSSP was originally scoped as a 'digital chain' for urban planning and regeneration with partners from academia and industry.[81] While it was the aim

[80] Christopher Alexander proposed the concept of 'Systems Generating Systems' as a first-order cybernetics approach but as a prescriptive rule-based model akin to KbD (Alexander 1968).
[81] Eight partners: University of East London (UEL): CECA, GIS and MA Alternative Urbanism; Aedas|R&D CDR; London Borough planners of Tower Hamlets and Newham; Planning consultants Urban Initiatives and 4M group.

of UEL CECA's Paul Coates to build a fully integrated planning chain to resolve explicit design stages, reminiscent of Alexander's *Systems Generating Systems*, CDR aimed to find fundamental yet implicit spatial planning heuristics from planning partners. Two sets of workshops were held, one with the planning consultants who implement planning guidance into projects through design using heuristics. And one with planning officers who judge configurations by design objectives and KPIs. The planning simulation prototype was to correlate urban design heuristics to algorithmic meta-heuristics evaluated by planning objectives. The declared aim was not to emulate design operations and appearances of existing deliverable as done by most urban planning simulations. That would defeat the purpose of the epistemic extensions provided by algorithmic design if existing knowledge was simply repeated (Derix 2012).

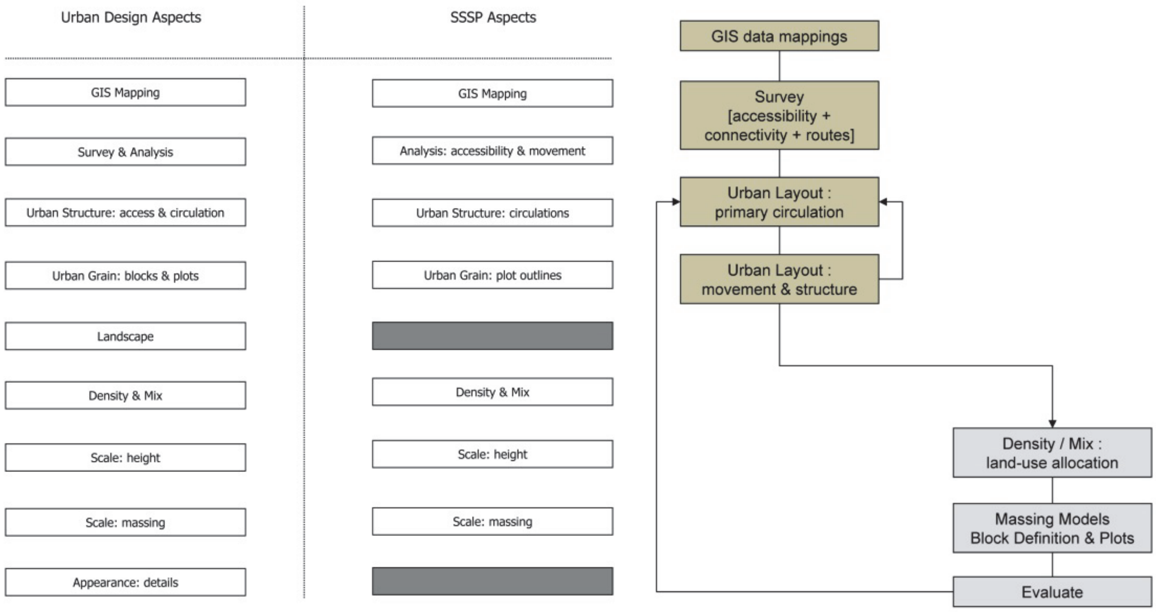


***Fig300**. SSSP, 2008: (left) diagram of the urban planning aspects as proposed by ByDesign of CABE and aspects dealt with by SSSP (CABE, 2001); and (right) the eventual basic achievable workflow as developed via SSSP*

Initially, a first phase prototype was developed that integrated several aspects of urban form into an aggregate monolithic design application for urban form. This appeared the most reasonable approach, designed on the linear rule-based generation principle. The Urban Block Editor prototype was considered by both public and private planners as akin to superficial workflow automation. And it was found that the generated massing states would not enhance their knowledge about the site and programme. Nor was it flexible enough to change heuristic approximations if the weighting of drivers or objectives were to change. Additionally, both parties rejected resulting configurations when formal masterplan appearances seemed to emulate known representations. In an integrated system where many aspects of urban form were weighted 'behind the scenes' concurrently, correlations between decisions and processes could not be visualized and observers (here planners) can either trust results when comparable to their known schemata, or reject them by suspecting faulty assumptions about operational or qualitative associations (Derix 2010).

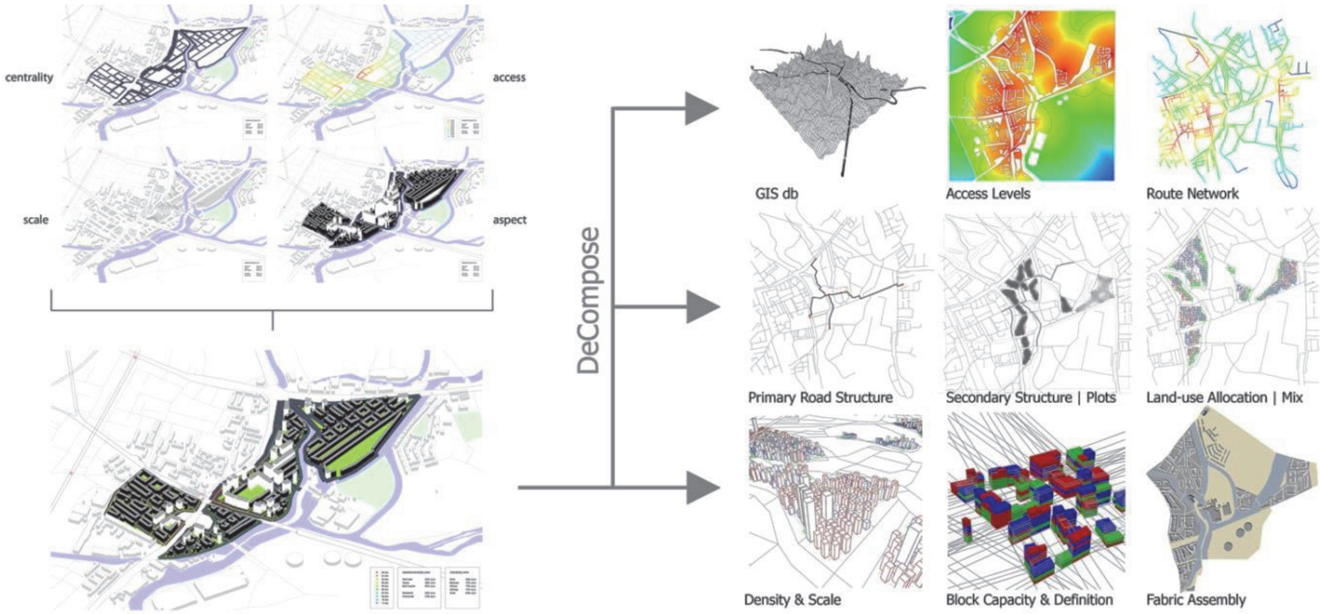


***Fig301**. SSSP, 2008: (right) the Urban Block Editor described in 7.1, integrating many aspects of urban form and performance into a single monolithic application (bottom left); and the resulting second prototype decomposing the Urban Block Editor into stand-alone meta-heuristic components within the planning simulation system (right)*

For the second iteration of the planning simulation, it was decided to work on the basis of parsimonious planning heuristics as stand-alone components without a hardcoded hierarchy. It was expected that this would allow *designing* planning consultants to alter one heuristic without having to decline the entire system, while planning officers could adjust design objectives. The integrated Urban Block Editor was opened into a series of design aspects correlating to a planning heuristic specified by planning guidance and our partners methodology for which a correlational algorithmic meta-heuristic was defined. From a single monolithic application, an open system of 7.5 interactive real-time applications resulted (.5 as one planning application contained two representations). The system could be executed as a chain like originally intended, including two manual stages, or organized into any bespoke workflow subject to brief, site and objectives.

Further advantages (see 9.1) emerged such as the visualization of behavioral processes of the meta-heuristic algorithms, which provides intuitive alignment with analogue heuristics without the observer having to know operational assumptions (briefly discussed 7.1). Also the open organization allowed for designers to elaborate individual stages manually or replace whole components when design heuristics already exist that have been developed into well formalized and generalized procedures. Such a step represents urban block typology definitions of lower density developments for example. Hence, it was decided to omit the encoding of this design stage.[82]

[82] The distinction into large high-density masterplans and smaller low-density masterplans when dealing with block typology definition had been confirmed on a smaller project for 3DReid in 2010 where this stage was omitted. However, on a larger project for Aedas (see EXPO 17 Village below), this stage was implemented and perceived valuable because little knowledge about the context's conventional urban block massing existed.

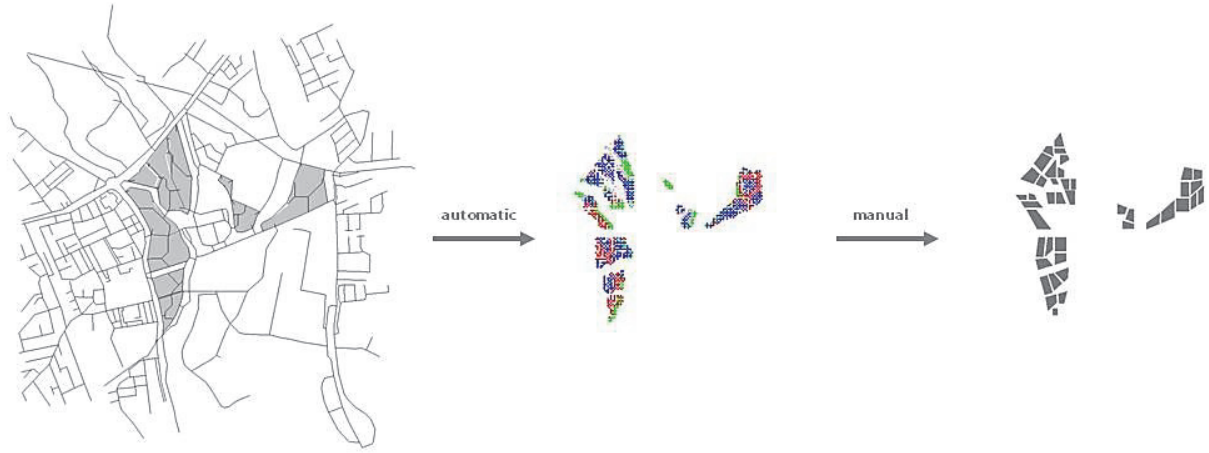


Fig302. SSSP, 2008: (left) the buildable area subdivisions output from the Urban Layout components (primary circulation and movement structure), followed by a possible automatic transition into a meta-heuristic for distribution of land-uses specifying mix& density configurations; at this stage, an analogue manual heuristic is suggested to outline the developable plot boundaries whose massing can either be automatically generated or manually drawn up in the next stage

A pilot masterplan was generated in 1.5 days from the new prototype and both the open planning system and produced massing masterplan presented in a concluding workshop. The second prototype was received very well by all partners who expressed a desire to work with the system and help refine it[83]. Partners also pointed to the advantage of the flexible workflow yet interoperable I/O of the meta-heuristic design components (such as access level, routing or mix & density simulations), because it was found that such behavioral occupant-centric heuristics relative to spatial qualities are not workflow-stage dependent but recursively applied across scales. Eventually, it was suggested that traditional deliverables should be revisited based on SSSP to offer new qualitative and quantitatively more complex information at different stages within the workflow (see 9.3).

The SSSP planning system definition was refined at Aedas and serviced as the Digital Masterplanning platform from which as series of projects were generated, amongst others the MIST 340 neighbourhood for Masdar Zero Carbon City 2009, a study for DavisLangdon in 2009, support for the Aedas masterplan of the former Kiew airport in 2009, the Aedas masterplan for Fort Halstead village in Kent, 2010, the Whitechapel CrossRail station urban impact study for TfL in 2010, a feasibility study for the Aedas Noida masterplan in 2010, the Barclay Bike journey visualization or Euston Crossing for TfL in 2013.

8.3.3 <u>The Open Framework for Spatial Simulation Framework (OFSS)</u>

The multitude of projects that the Digital Masterplanning components were applied to facilitated the generalization of core heuristics into meta-heuristics or the refinement of existing meta-heuristics. By 2010 an outline framework including meta-heuristics and algorithms for design aspects other than urban design such as described in previous chapters, started to consolidate. Agreed formalisms,

---

83 Public planning officials pointed out that they did not have the funding to implement the system while Urban Initiatives and 4M continued to collaborate with CDR.

generalizations and interoperability became vital to the work of CDR as two geographically separated teams needed to work on common projects and exchange source code. The geographic split also enforced a division of labour into fast project implementation and slow project development. One team integrated into architects' workflows in the office, more often applying and testing developments on live projects, while a satellite team developed new algorithms, generalized project heuristics and compiled the source code framework. On long-term R&D projects both teams collaborate on new algorithms and meta-heuristics and compile resulting generalized parts or components into the framework. Concise single-person scoped projects could be developed by both the satellite and office-based teams. Eventually, a robust framework evolved that is shared between both teams and serves as the basis for all projects.

Before illustrating how a project is compiled from the OFSS, a short description of the framework and its components is provided. The OFSS is programmed in the Java language and written, compiled and built in an Integrated Development Environment (IDE) called Eclipse[84]. The Java language represents an object-oriented programming language of the *structured imperative* programming paradigm[85]. Initially developed by Sun Microsystems and now maintained by Oracle, it is currently in version 8. CDR has programmed in the Java language since 2007, before which mainly Visual Basic, C and C++ were used.

The OFSS currently contains 14 integrated components[86]. Components represent a family of code that constitute some sub-system of the open design system whose representations and behaviours can be connected to create more complex models. The 14 components can be differentiated into three types: functional, spatial and geometric. Spatial components contain both generative and analytical representations and algorithms for configuration and occupancy. Integrated components are fully generalized, so that their object classes are abstracted to such as degree that they can be instantiated directly from another component without having to adapt the class' structure or data types, leading to high interoperability (Miranda and Derix 2009). The framework diagram in Fig303 shows the 14 connectable components. The top row shows components of functional and geometric families and the bottom row shows spatial families. Each component contains a structure of representations of which a selection is shown as dependency nodes. Those representations call the actual algorithms of which again only a selection is shown for clarity. The node and sub-node labels have been summarized for legibility.

[84] https://eclipse.org, accessed 27.06.2015
[85] All procedural languages are *imperative*, instructing the computer what operations to execute. Most imperative languages are structured instead of un-structured or simply flowing linearly like the original Basic language (not Visual Basic). Object-oriented programming (OOP) is an evolved modular version of structured imperative programming allowing for generalized abstract object classes and their instantiations into specific contexts. Most languages used in architectural design nowadays are based on OOP.
[86] Coincidentally, the System of Systems research components seem to align well with the components of the OFSS (see http://en.wikipedia.org/wiki/System_of_systems).

Functional components include

1. Framework manager
   initializing, OpenGL (graphics libraries) interface, event handling, renderers, protocols to OS etc
2. Graphical User Interface
   elements for visual explanation and operation in application window
3. Interaction
   methods for interaction such as geometry selection and external input/output device adapters
4. Collections
   array elements such as lists and maps (2d lists) etc
5. Input/ Output
   Data formatting for file types such as DXF for CAD or SVG for vector images or CSV for numeric tables etc

The key geometry component includes

6. Geometry
   primitives (all geometric basic elements such as points, lines, surfaces etc), transformation matrices, tesselators for rendering and operations (such as Booleans, intersections etc)
7. Mesh
   meshes for arrayed surfaces modelling such as building and city models; but also meshes as grids for spatial calculations
   - ➔ *the mesh components is a hybrid and also contains some topological and Euclidean analysis for spatial components*

<u>Refers:</u> 6.1, 6.2, 7.1-3

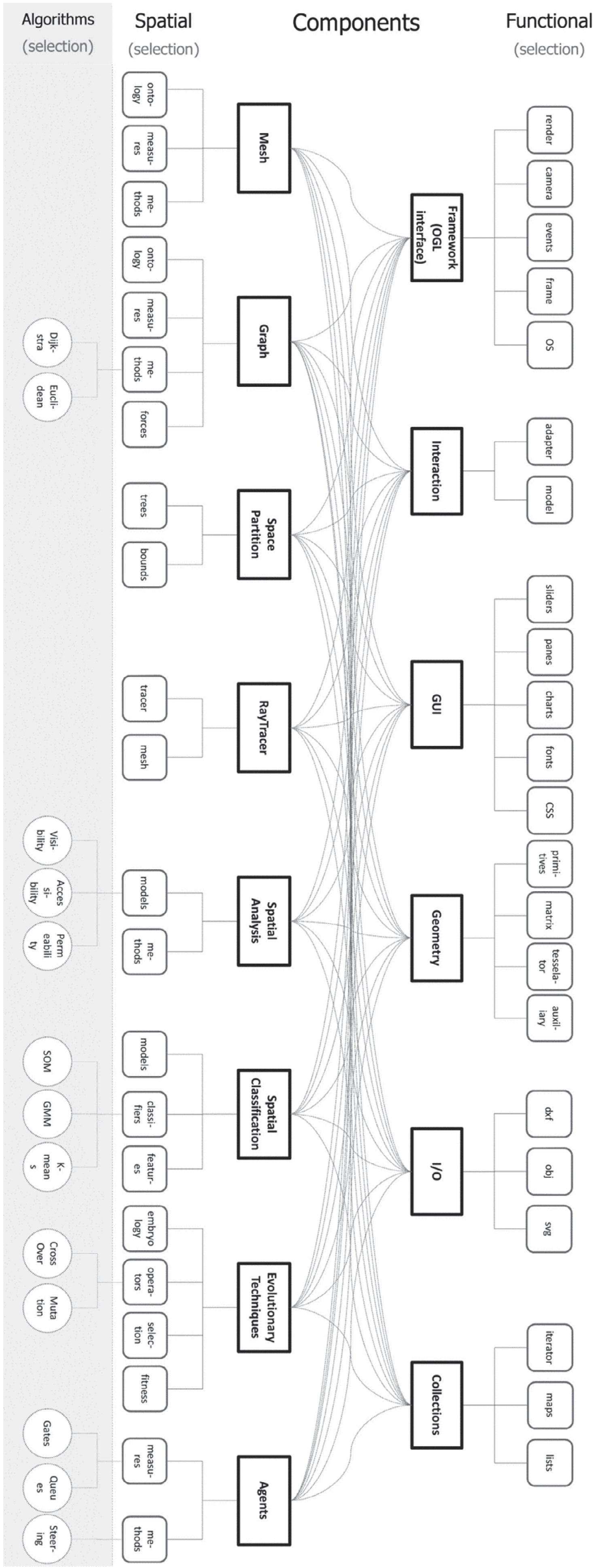


***Fig303**. OFSS schematic diagram, 2013 (rotated): the framework design and developed in the programming language Java comprises 14 components, including 5 functional, 1.5 geometric and 7.5 spatial (0.5 because they share tasks); for visual clarity the diagram does not show all sub-components and algorithms*

Spatial components include

8. Graph

    extracts and builds graph representations from points, edge networks and meshes and contains graph/network theoretical measures; methods and models contain the core calculations for spatial analysis models such as visibility, accessibility and permeability algorithms, force-directed graphs

    Refers: 6.2, 7.1-3, 8.1, 8.2

9. Space Partition

    creates space partition structures for faster searchers, including kd-trees; often used with visibility analysis to pre-order spatial topologies of surfaces; but can be used for many algorithms such as tree-maps

    Refers: 6.2, 7.1-3

10. Raytracer

    ray-tracing, using also Space Partition and Mesh

    Refers: 7.1, 7.2

11. Spatial Analysis

    serves as the algorithmic interface to call many algorithmic procedures such as visibility analysis methods including VPTA, access methods, route calculations, media axes etc; the calls connect to other spatial components like Mesh, Graph etc

    Refers: 6.1, 7.1, 7.2, 8.2

12. Spatial Classification

    representations and algorithms for classification of spatial properties including the SOM algorithms and other classifiers; also requires feature representations such as binary or real feature vectors and their comparison methods

    Refers: 6.3, 7.3

13. Evolutionary Techniques

    representations of embryologies, evolutionary operators, selection processes such as standard competition methods like roulette wheel or archiving methods like Pareto optimization, annealing algorithms

    Refers: 5.1, 7.1

14. Agents

    multi-agent system for simulations such as People Movement or some layout models such as ADEC; contains behavioural algorithms like Reynolds' 'steering' methods

    Refers: 5.2, 6.1, 6.2, 7.2, 8.1

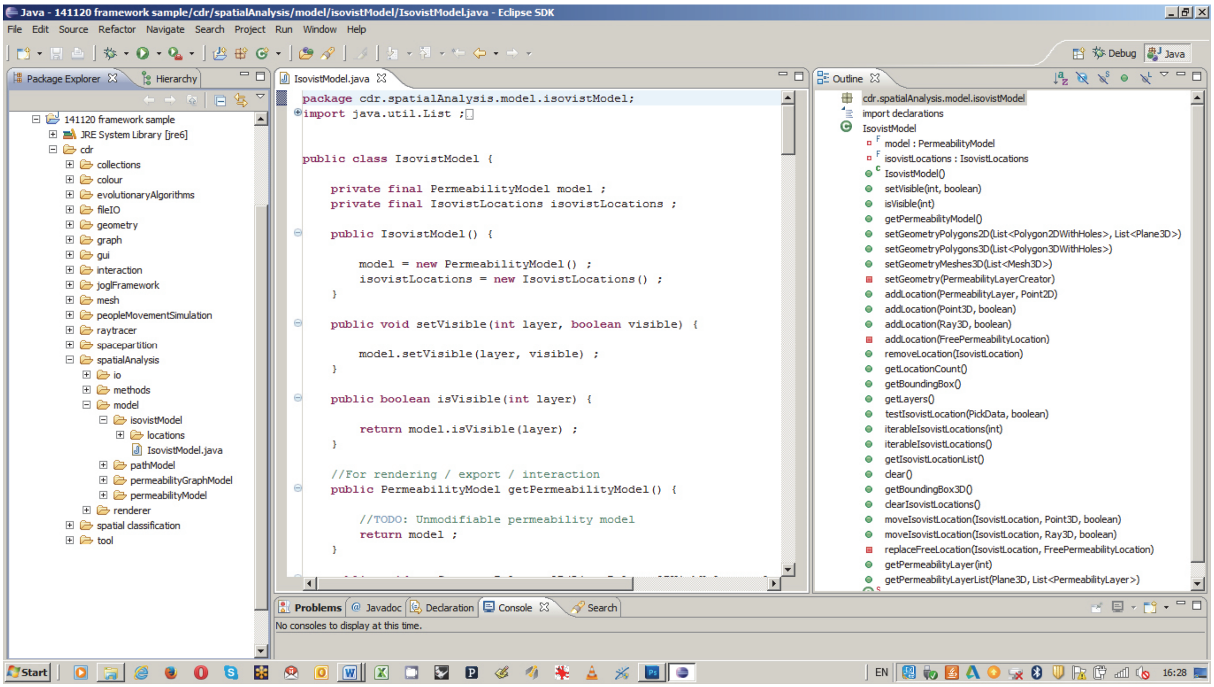


Fig304*. OFSS, 2013: the framework in the programming interface of the Eclipse Software Development Kit (SDK), showing the system components (left window), one code module (IsovistModel) and its structure (right)*

Most components with their representations and algorithms are not explicitly linked but their generalized abstract object classes allows for flexible *ad hoc* association. Some components work as interfaces like Spatial Analysis where the representation and general algorithmic structure relies on further calculations based on the representation of a particular algorithmic model. For example, the route models of chapter 6.2 form an associative system (calling queues) between many geometric and spatial components (all systems use most functional components) such as

Spatial Analysis.Model.Path <> Graph.Paths <> Mesh.MeshGeometricMeasures

Apart from the components and their dependencies built by CDR and SUPERSPACE over 10 years, a small selection of external libraries is employed for some specific tasks. The most common and general of those tasks is the OpenGL graphics library, which is interfaced directly from the Framework component. As discussed, ray-tracing employs elements of the Sunflow library (6.1) or the RIBS 3D spatial topology representation (6.3) employs the digital physics library Traer 3.0 for spring systems[87]. Other supplementary external libraries have been and are occasionally in use and for legal reasons it is important to point out the use of external libraries via the General Public Library convention (GPL) that regulates the licencing of open source software[88].

The Open Framework for Spatial Simulation won the 2010 Commendation for the Royal Institute of British Architects (RIBA)'s President's Medal for Outstanding Professional Practice-located Research. The jury commented as following: *"Many other Research Units into Computational Design might exist in the world, but this*

87 http://murderandcreate.com/physics/, accessed 10.12.2014
88 Managed by the Creative Commons organization: https://creativecommons.org, accessed 10.12.2014

one has a 'local' character and a particular story, which roots it in the English history of computation in architecture, and makes it both original and specific." RIBA awards panel, 2010.[89] *[…] As such, the work of the group could be regarded as a vernacular of architecture design computation."* (RIBA Journal Dec/Jan 2010/11, p52)

INSTANTIATIONS

As discussed in the introduction to this chapter, models presented as case studies previously represented concise stand-alone applications. Most of those applications reflected only one component of a larger system. Having introduced the OFSS, the whole system for two of those applications can be shown diagrammatically within their context as instantiations of the framework. A selection of three instances was done on the basis of three distinct types of associations interfaces. The three instances embody three types of scale, architectural space and meta-heuristics. This range of associations shows that the framework is not dominated by a specific ontological or epistemological representation, heuristic or component but enables instantiation of contextual systems from the abstract meta-system.

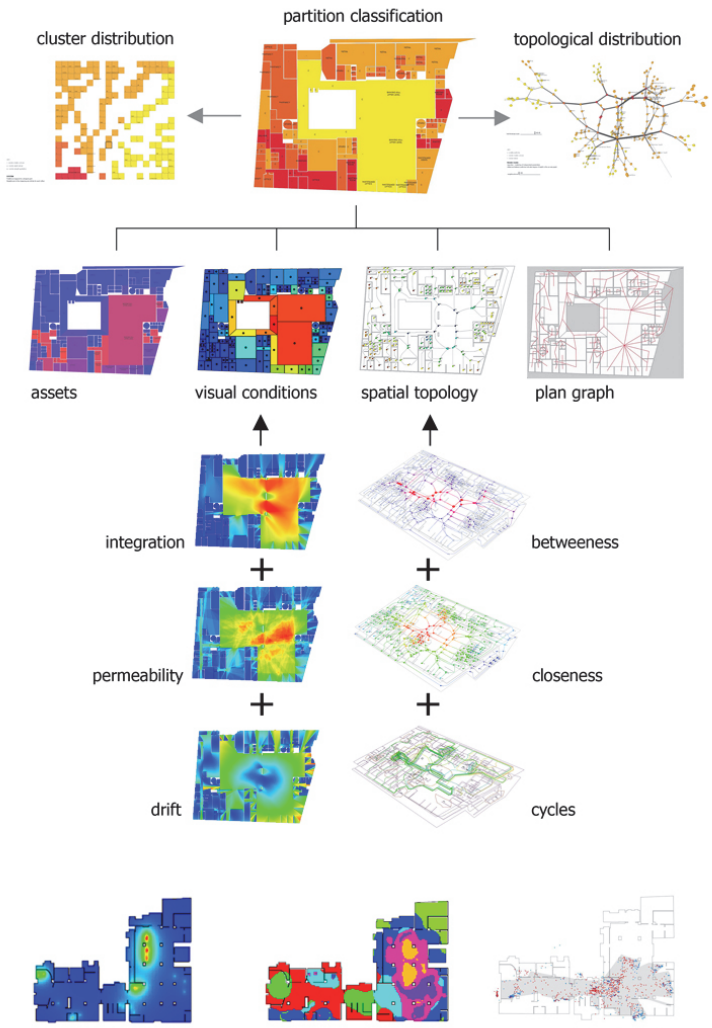


Fig305. *RIBS, 2010-3: all functional and geometric components of OFSS were used for the project; the 'Spatial Classification' component was responsible for associating other spatial analysis components*

[89] See http://www.architectnews.co.uk/aedas-win-riba-award-cms-1266

RIBS | Cognition Algorithm as Associative Structure

The first instance puts two models of the RIBS research project into context. The STG in chapter 6.2 Graphs – Behavioural Diagrams and the IAA in 6.3 Networks – Associative Fields dealt with behavioural affordances of a spatial configuration and the cognitive organization of spatial properties. RIBS was not meant to be a generative system but as described in chapter 7.3 *Meta-Cognitive Configuration Of Space* and the generalization of IAA (the SpaceProfiler), the organization of intuitive information into schemata provides generative concepts that help to identify a design strategy on the basis of which tactical decisions are made.

RIBS generated a vast amount of spatial and occupancy data and the IAA was developed to organize the data to reveal patterns of use performance (in this case for the resilience of the spatial infrastructure against terrorism). The core algorithm employed to generate spatial patterns associated to occupation, cognition and operations was the Kohonen self-organizing *feature* map (SOM). The SOM provides a cognitive model with a specific epistemology, namely the generation of schematic classes about a space by differential comparison (see 3.5 Hebb's competitive learning) of behavioural and cognitive performances. The SOM was however embedded in a larger spatial classification component that extracted data from analytical, generative and simply numerical feeds (such as video logs), generated by other components, that generated input data for mining.

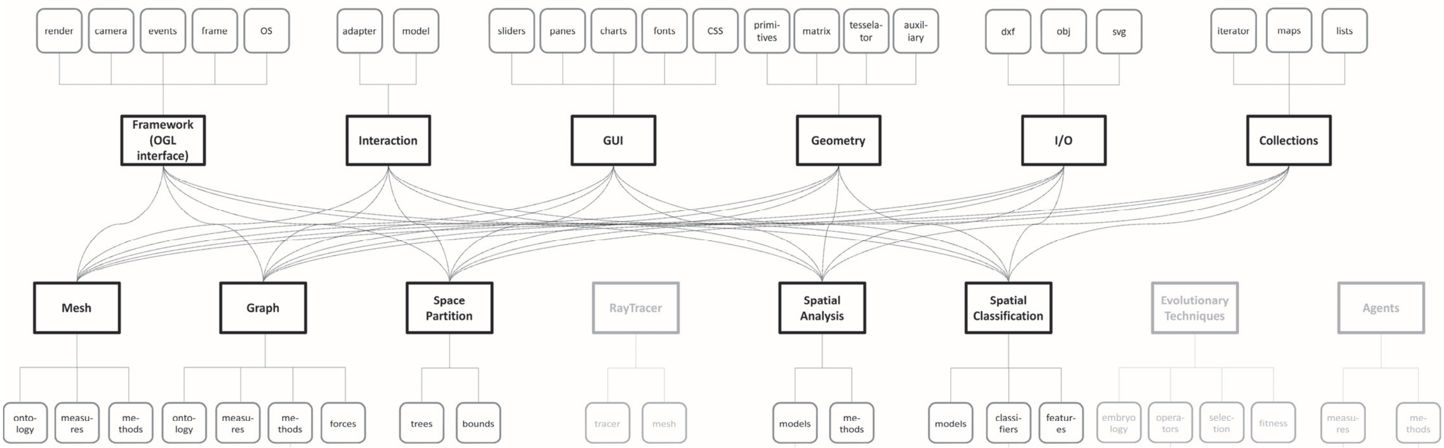


***Fig306**.: RIBS, 2010-3: eleven components were used for RIBS. All functional and geometric components; the 'Spatial Classification' component was responsible for associating other spatial analysis components*

RIBS therefore represents an associative structure with an epistemic rather than generative emphasis. Components reading space, generating data and building schematic representations can be executed as a chain but also operated within smaller groups. The observer mediating the components and their original input representations is learning about schemata and their associations within an open system that is not fixed to produce specific states (deliverables). Such a system is allocated during the strategic design phase of a design process to enable the detection of hidden connections within the design space. The observer learns about his assumptions about space and occupation as correlations between the two. They are revealed to help him empathize with the intended user of the space. RIBS instantiated a system of *empathic spatial knowing*.

FuCon | Behavioural Workflow as Associative Structure

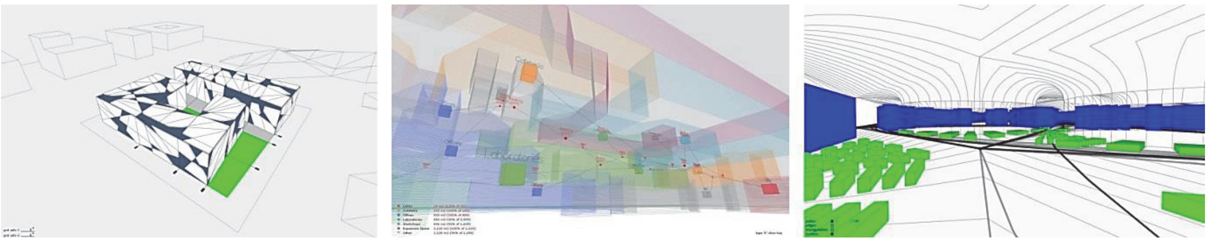


Fig307 *FuCon, 2010-11: three applications across three scales linking massing and envelope with capacity and programme allocation and floor layout with furniture grid. The configurations are designed simulating movement behaviours*

A second type of instance refers to the design of an open associative workflow akin to the *Floating Room* system but produced directly from the OFSS. Again two of three applications developed for the Future Construction (FuCon) project have been discussed as FuCon3 in chapter 6.2 and as FuCon2 in chapter 7.2. The third application dealt with the massing and envelope configuration within which the latter two are situated. This first application uses an *arrangement of lines* algorithm (O'Rourke 1994, p199) that partitions a polygon into convex shapes to which programme and density can be attributed. The same algorithm was used to generate a draft façade pattern in 3D. The three applications were digitally linked through the virtual reality platform VRfx of the Fraunhofer Institute (Krause et al. 2011). The observer incorporated both roles of designer and user simultaneously: he could manipulate the design drivers of the building geometry on three scales – envelope and mass, building configuration and floor layout – and could at the same time evaluate the configuration by spatial immersion. The tracking system allowed the observer to virtually move through the circulation spaces during runtime to calibrate design decisions with perceived occupancy affordances. The three scale-based design simulations could be selected and used in any sequence via a hand-held tablet computer.

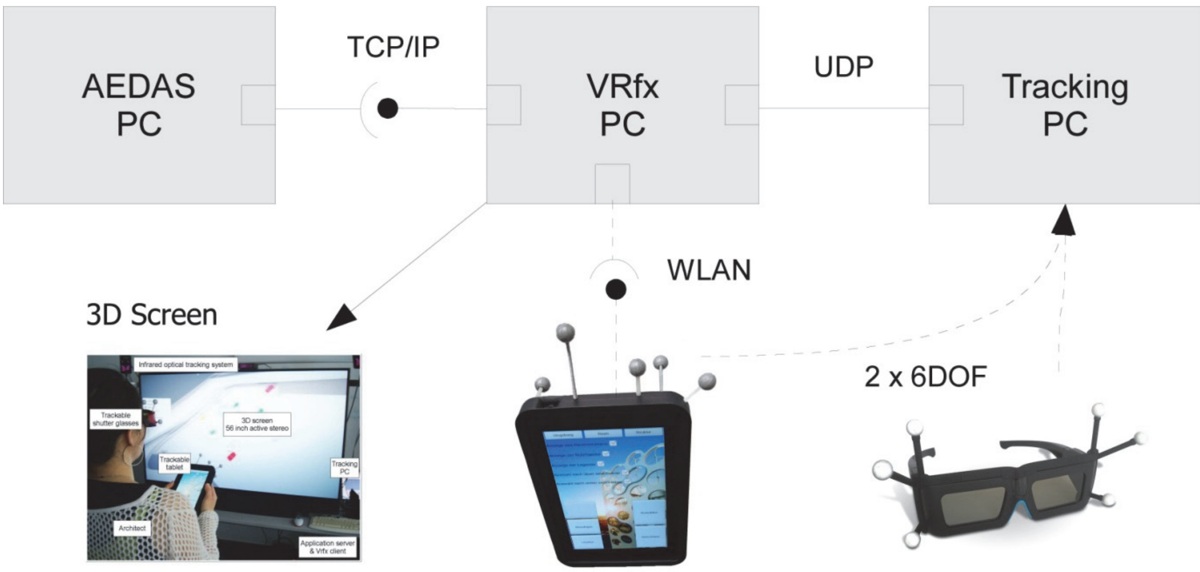


Fig308. *FuCon, 2010-11: the combined FuCon demonstrator system was composed of the OFSS linked to the Fraunhofer VRfx system; a hand-held tablet computer was used to drive the algorithmic simulation from the OFSS; the tablet and 3D goggles were tracked to situate the observer as design and user inside the spatial configurations (Krause et al. 2011)*

***Fig309**. FuCon, 2010-11: immersive demonstrator set-up and in use at the BAU 2011 expo in Munich, Germany*

The principal design concept behind FuCon was not limited to its technical specification as a connected system between the OFSS and VRfx. On the contrary, algorithms were selected to demonstrate the ability to design *inside-out* through computational systems, while also encapsulating a fundamental design heuristic: recursive spatial pattern across scales. The spatial immersion via virtual reality was to support the notion of situated *inside-out* planning (Krause et al. 2011). The recursion of patterns and self-referential structures across scales was intended to aid orientation of the occupant at different scales and locations and has been adopted by many architects such as Frank Lloyd Wright and Mies van der Rohe (Frampton, 1985). This aspect of form and cognition is usually discussed in context of Gestalt psychology and refers to the perception of objects and their integrity or scale and symmetry (Arnheim 1974). The nesting of the edge-bundling algorithms across scales and the line arrangement partitions follow this concept algorithmically. The algorithms are meant to simulate occupancy behaviours and thus the scalar proportions support orientation by identifying with movement patterns rather than formal elements. FuCon represents an open design system that enables the observer as designer and user to develop spatial configurations from *inside-out* by implicitly associating behavioral patterns through situated actions. Those can be evaluated during generation by virtual cognitive use.

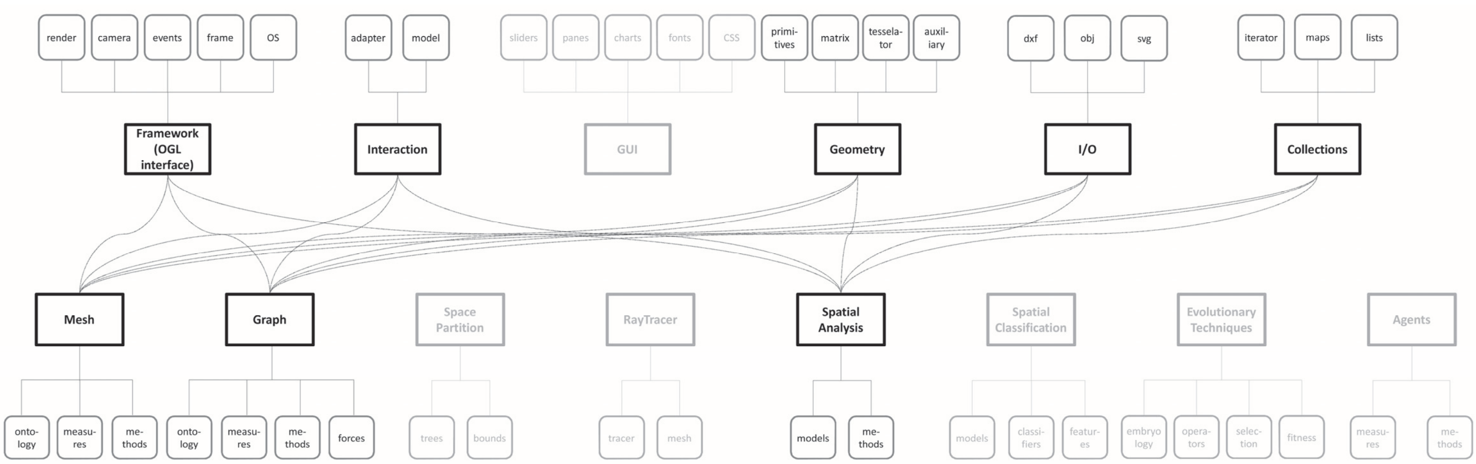


***Fig310**. FuCon, 2010-11: eight components were used to develop the open system. The functional 'framework' component also linked via a formatting protocol the OFSS with Fraunhofer's VRfx. Hence, no GUI component was needed as all interfacing was handled by VRfx*

EXPO 17 | Spatial Topology as Associative Structure

The third type of instance represents a project-specific workflow. Aedas entered the invited competition for the World EXPO 2017 Village, named Green Block, in June 2013. Two stages of the application of OFSS need to be distinguished: the competition phase and the concept planning phase. The entry won First Prize and subsequently, the workflow was adjusted.[90] For this instance type a hybrid workflow bespoke to the masterplanning design is briefly discussed.

***Fig311**. EXPO 17, Green Block, Aedas, 2013: rendering of the masterplan submitted for competition entry*

A computational workflow for the competition was developed that would also generate the urban morphology. The concept was derived from a previous village masterplan in 2010: leaf-like cell aggregations differentiating plot sizes by access patterns and mix. The generic workflow works as following:

- Agent-based aggregation of programme
  development quantum assembled according to adjacencies and site-feature proximities; generating land-use allocation and phasing strategy

***Fig312**. EXPO 17, Green Block, Aedas, 2013: agent-based land-use aggregation; attractors are interactively seeded and programme inserts itself and aggregates by adjacencies*

---

[90] The masterplan just gained planning permission for the $370 million development: http://ahr-global.com/Expo-Village-Masterplan-Green-Block, accessed 21.12.2014; after many iterations, it does not resemble the competition entry any longer

- <u>Graph-growth for primary road network</u>
  land-use nodes seeded by the agent-based application are connected into tree-graphs using the biological concept of *hyphal growth*[91]

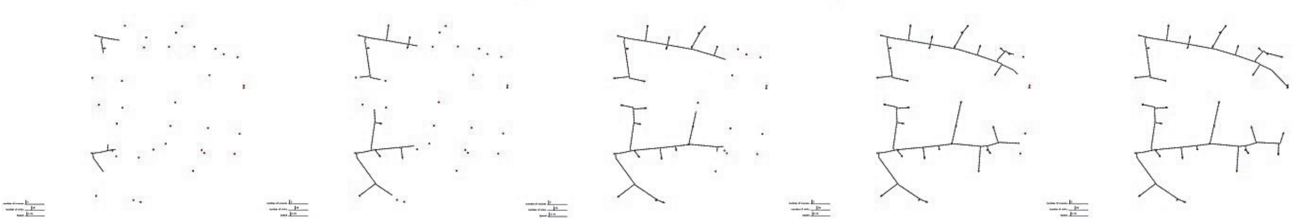


Fig313. *EXPO 17, Green Block, Aedas, 2013: graph growth connecting land-use seed points*

- <u>Recursive grid definition</u>
  inserted vertices from the graph growth process recursively interpolate new nodes into the hexagonal grid to refine the tissue and produce plot outlines

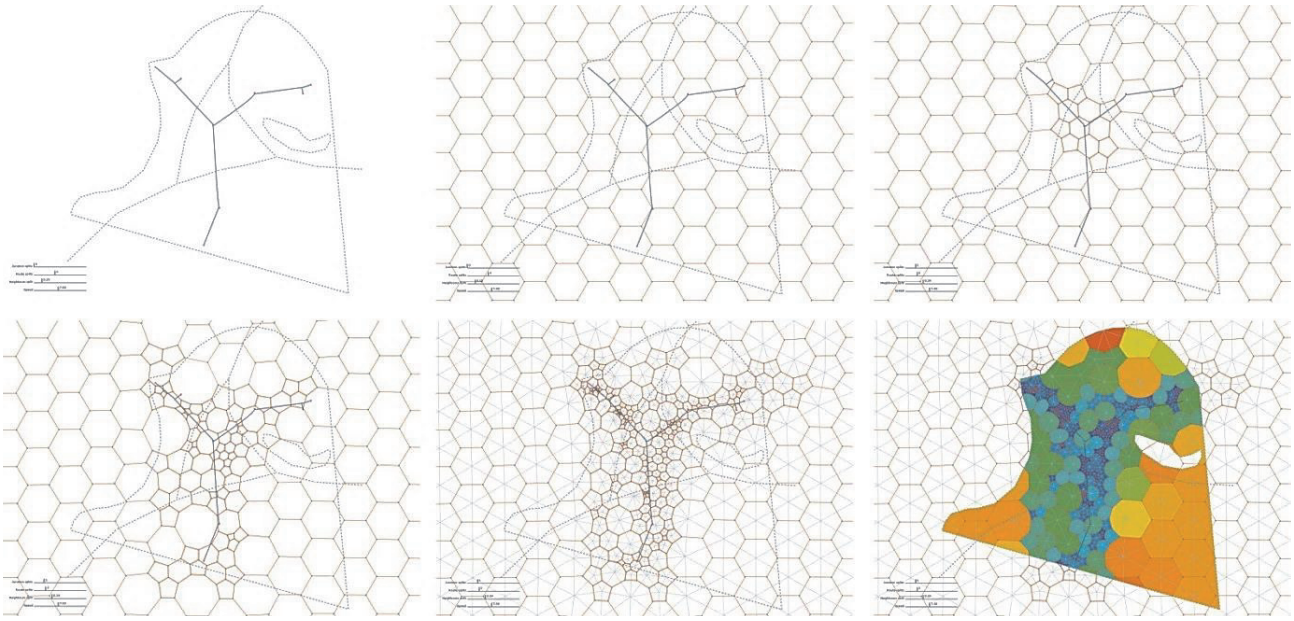


Fig314. *EXPO 17, Green Block, Aedas, 2013: the grown graph vertices interpolate new nodes on the grid and recursively subdivided it to arrive at a series of plot sizes adequate for the land-use footprints*

- <u>Topology analysis for public programme allocation and morphology</u>
  betweenness centrality is calculated for the grid edges and well as plot centres; the centrality / integration values by plot indicate where public functions (sports facilities, light-rail station and bio-domes) should be allocated within un-attributed plot cells; centrality values along edges rather than plot nodes provide first set of street-aspect ratios

- <u>Shortest-routes accessibility for secondary circulation</u>
  using Dijkstra routing algorithms to generate access levels for each building entrance to all public facilities; values generated along route

---

[91] Fungi grow via a hypha structure that extends from endpoints and branches dependent on environmental information: http://en.wikipedia.org/wiki/Hypha, accessed 22.12.2014

edges adapt street-aspect ratio values generated above, based on stringent fire-access regulations in the Russian planning standards called SNIP[92], which are also applied in Kazakhstan

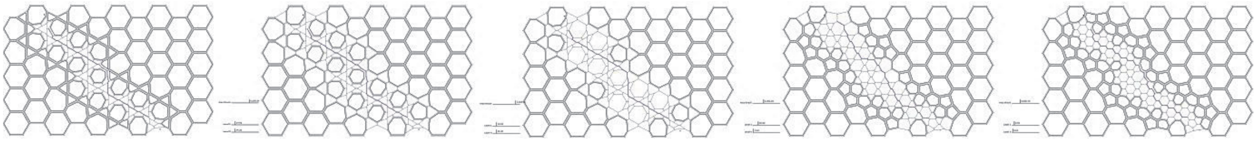


Fig315. *EXPO 17, Green Block, Aedas, 2013: five stages of refining the grid according to access routes to public facilities and fire regulations (spacing between plots for fire engine access and turning circles)*

- Urban block definition
  simple calculation of plot areas determine the block typology; depending on plot aspect ratios and area, either a court-yard or terrace typology is inserted whose scale is dependent on the above street-aspect ratios and limits of the development quantum

- Ray-tracing for urban bock refinement
  using the ray-tracing, space partition and mesh components, all apartments are given an annual 'insolation' value, which must be over two hours sunlight/ day over the whole year

- Surface generation for bio-domes
  the geometric definition for the bio-domes was conducted using Rhino Grasshopper, based on a recursive subdivision algorithm developed by CDR member Anders Holden Deleuran[93]

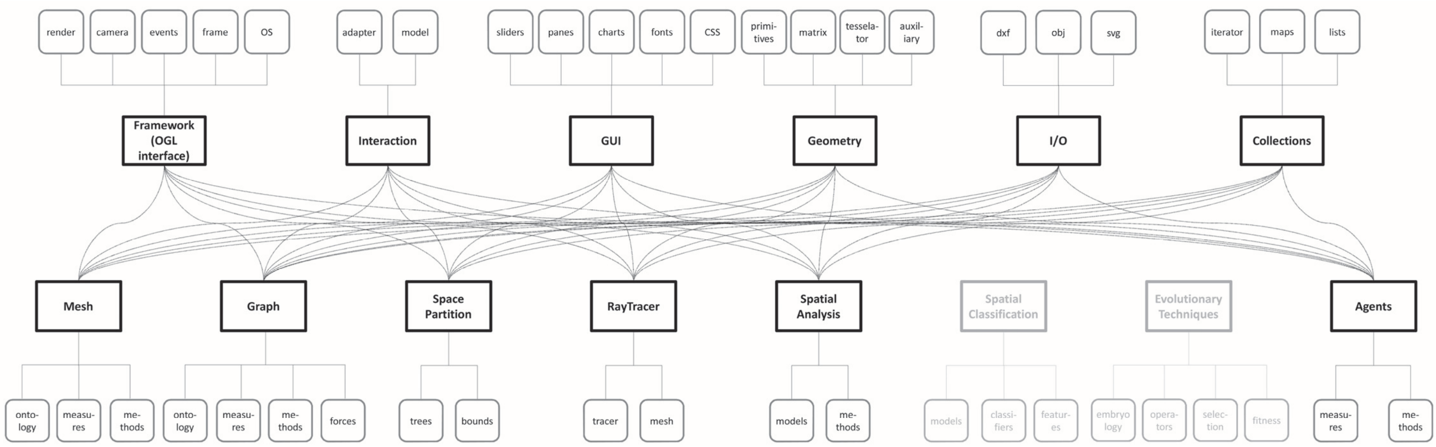


Fig316. *EXPO 17, Green Block, Aedas, 2013: 12 components of the OFSS were used to assemble the workflow*

All workflow stages are built via the OFSS bar the last geometry definition. The competition workflow included only the five stages, 1-4 and 8; stages 5-7 were added during the first concept planning phase. The associative structure across workflow stages 1-5 is based on topological properties. Each stage and scale increases the topological associations between different morphological parts. Any

[92] http://snip.com, accessed 22.12.2014
[93] http://www.andersholdendeleuran.com, accessed 22.12.2014

current design state of the urban configuration effectively constitutes a form of consensual domain between topological and morphological schemata.

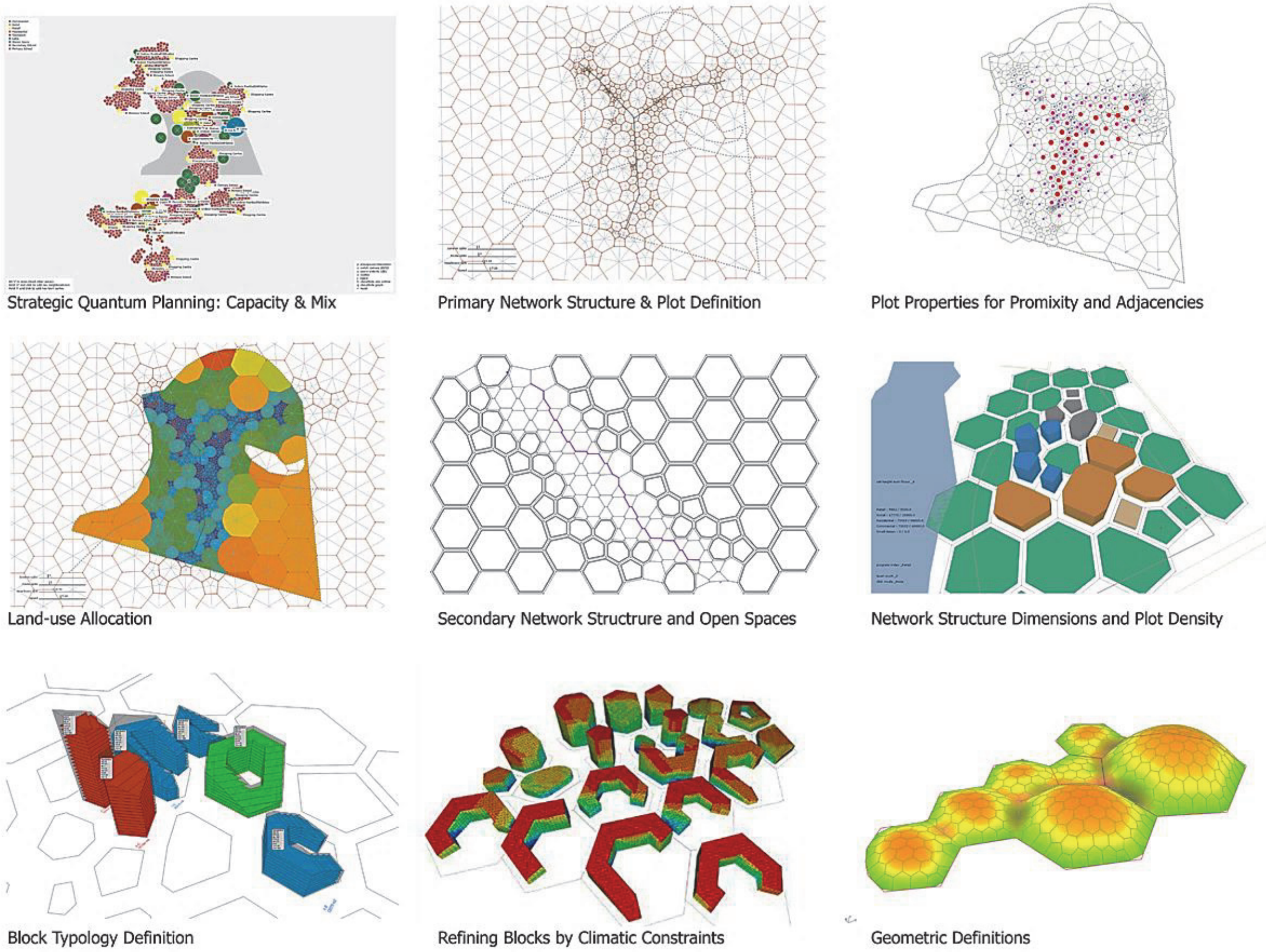


***Fig317**. EXPO 17, Green Block, Aedas, 2013: nine workflow stages assembled from the OFSS*

This masterplanning instance resembles the prescriptive workflow of *Floating Room* and the urban typology can in fact by re-generated by observers other as well as the original design team. The observer-designer becomes an equivalent part in an autonomous system. The competition entry was in fact not presented through visual boards as standard demands but via recordings of the workflow dynamics, which might have swayed the decision by the client in favour of an efficient flexible design system, since three alternative sites required testing.

8.4 CONCLUSIONS

The discussion of Field Organizations demonstrated that more generic representations and structures exist allowing for mediation between heterogeneous actors: the human observer, the simulated user with behavioural performances and cognitive affordances, and spatial configurations as geometries. Properties and performances of those actors are encoded as systems into computational components. Those in turn are generalized and compiled into an abstracted meta-system from which contextual instances can be created. The equilibrium between autonomous components is seen as an important quality of Field Organizations because only as structurally autonomous actors can they interact with other components (as systems) to generate associations between their knowledge domains. When this happens, two or more components are defined to be *structurally coupling* and producing new knowledge, called the consensual domain. This theoretical description might sound abstract in contrast to the general discipline of computational design in architecture where the main thrust of R&D has been about the efficient automation of existing and standard knowledge. The OFSS as a meta-system has managed to provide a computational basis from which to compile design systems built on the correlation between spatial configuration and human-centric affordances. This is particularly difficult within a professional context when design workflows are time-restrictive and generally driven by non-human centric efficiencies for which most other computational design systems exist.

8.4.1 Structures

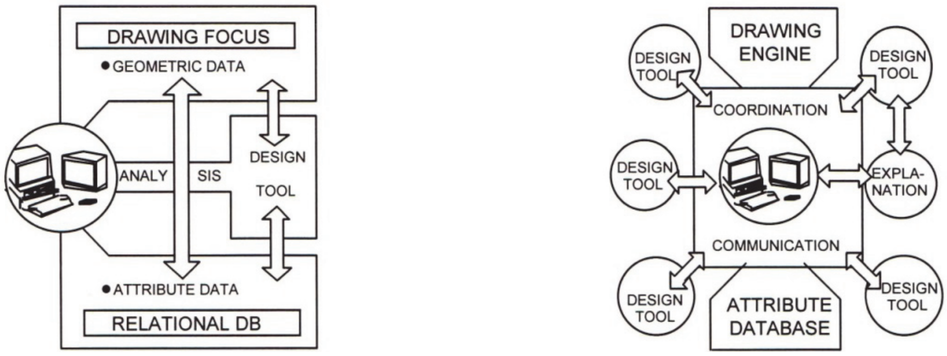


Fig318. *Computer-based Design System (Pohl and Myers 1993): from a single agent (left) to a multi agent design system*

The Knowledge-based-Design discipline (KbD) has explored structures of the design process and hopes to distil them into computational models. A well-known model is John Gero and Michael Rosenman's Function-Behaviour-Structure system (Rosenman Gero and Maher 1994), which attempts to explicitly encode all possible ontological elements and their parametric associations. Those expert systems started to be perceived as exceedingly over-constrained and too monolithic to be applicable for an actual design process (Liggett 2000). Richard Coyne (1990) proposed a radical new approach in 1990 by introducing the notion of connectionist models where parametric and knowledge schemata would not have to be specified *a priori*. But the connectionist concept model remained theoretical apart from Petrovic's draft generative PPD-AAM model (Petrovic and Svetel 1993). Less deterministic structural diagrams of computational design systems were proposed by KbD members like Jens Pohl and Leonard Myers (1993). Criticizing the over-constrained model of Gero and Rosenman, their system diagrams pointed towards the decomposition of the computational model into several components with their own logical structure, albeit not necessarily with an autonomous self-organizing epistemology as proposed by the New Epistemologists. The OFSS manages to blend both requirements into one framework, namely autonomous components based on self-organizing algorithms that can be associated without over-constraining the observer and hence not suppressing his creative routines. To use Schön's words (1983), a *reflective conversation* opens up in which narratives can emerge based on novel schemata that integrate all observers.

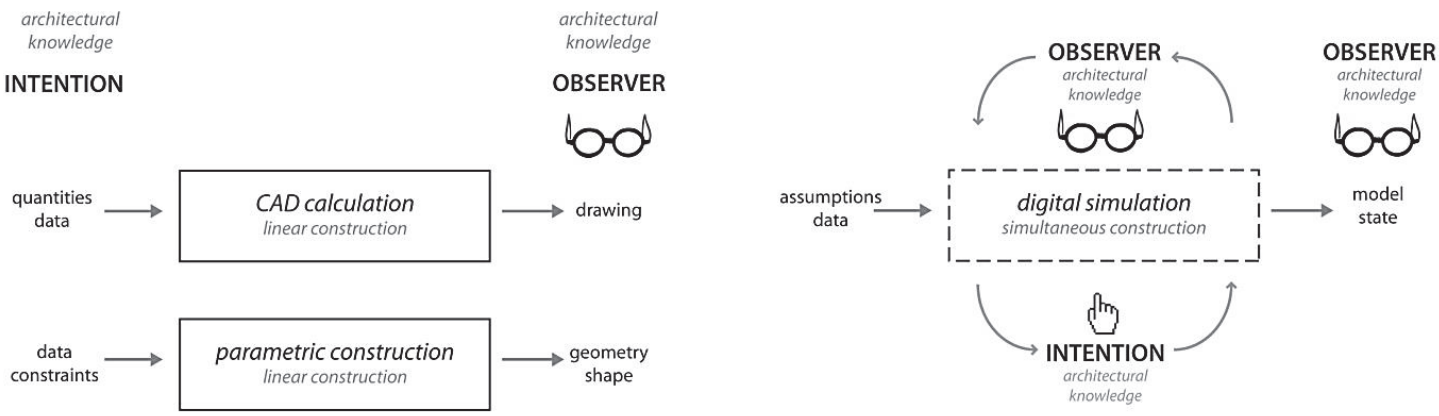


Fig319. *Observer knowledge: from linear to open and cyclic associations reminiscent of von Foerster's 'second-order cybernetics' (slides from the author's RIBA R&D Awards ceremony presentation, 2010)*

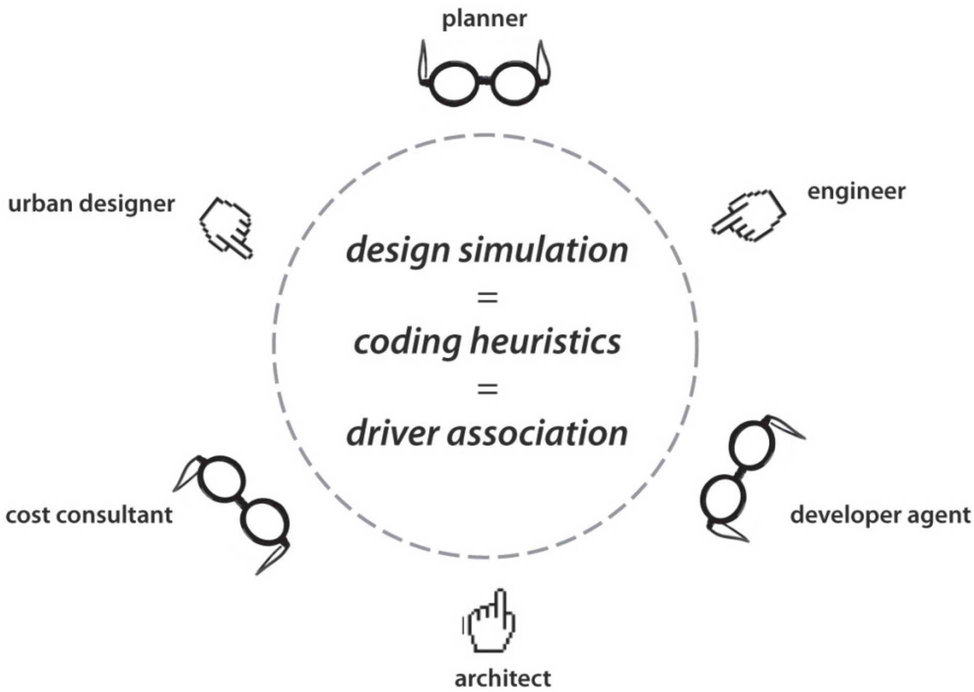


Fig320. *All stakeholders's intentions can be absorbed through the meta-heuristic visualization of coupled processes, generating 'visual narratives' that allow for common identification with the system without recourse to sector-specific terminology (slides from the author's RIBA R&D Awards ceremony presentation, 2010)*

8.4.2 Observer | Agency

The OFSS is neutral to the epistemology of one specific actor. While each components itself represents some form of agency of the observer or of the algorithm as discussed in previous chapters, the meta-system itself is not weighted towards a specific schema or intention. In general, it could be said that the meta-system is biased towards the *designing* of human-centric spatial configurations as opposed to structural or geometric aspects of building form.

The instances discussed in this chapter including the two case studies have demonstrated that the meta-system is independent of an *a priori* agency, intention, typology or scale, including the observer. This differs widely from the KbD research into *prototypes* and *design cases*, which were meant to be compiled as archived reasoning of specific typologies (ontologies of a schema) (Gero and Maher 1993).

Instead, three orders of abstractions for the observer and the field have been illustrated that structured the chapter threads:

- Remote observer agency with algorithmic epistemology
  ➔ instance three: EXPO 17 Village workflow
- Situated observer with limited epistemic agency
  ➔ instance two: FuCon immersive generative system and SSSP
- Learning observer with emerging schemata
  ➔ instance one: RIBS spatial classification for occupancy affordances

The transition from simulating a) observers' externalized heuristics to b) mediating observers' implicit empirical meta-heuristics to c) learning about the observers' internal cognition has been documented. A shift is possible from simply simulating the 'user out there' to the 'user in here'.

8.4.3 Field – Empirical Typologies

Finally, the concept of *field* as used in the heading of this chapter is as stated in the introduction to this chapter an extension of Allen's mono-dimensional graphic organization into an organization of a multi-dimensional system. This system as meta-system can be instantiated into a large variety of knowledge domains and in that case simply represents a *network of consequences* (Latour 1999), i.e. a single perceived field of actions. But unlike Hillier's *laws of the field* where only a single action correlated a single state, a more heterogeneous range of heuristics, meta-heuristics and cognitive affordances can be associated simultaneously in an open system. The field represents the *field of associations* between behavioural and cognitive affordances in relation to user-centric spatial configurations. No ontologies or parametric procedures are designed-in to reflect specific architectural typologies. Only a network of associated processes is provided whose abstracted units are associated by events. The lack of architectural typology is replaced by the focus on new *consensual domains* that propose *generic functions* or scenarios of use. The association by events is driven by empirical knowledge regenerated by instances of the meta-system (as clearly shown in *Floating Room)* and thus the resulting states of spatial configurations embody empirical typologies. A typology is an open field correlating experiences of behaviour, cognition and spatial configurations, generating architectural geometries from occupancy as suggested by Hillier and Leaman's *manifolds* (1974) and Steadman's description of the *generic dissection* (1983). In this *anthropo-spatial* meta-system, the observer designs with implicit empirical benchmarks of user-spaces (Derix 2014).

9 CONCLUSIONS

The final chapter first reviews the levels of achieving the initial objectives set out in chapter one. The objectives are briefly discussed in the context of the four development chapters five to eight and the resulting OFSS. Further, the proposed framework approach - beyond the OFSS generally called USOM - is examined from two perspectives: the impact of human-centric design computation on professional practice and conceptual insights stemming from the developments. Eventually, a concluding statement expresses an aspiration for the architectural field resulting from this thesis.

9.1 OBJECTIVE 1

Correlate human-centric performances to aspects of spatial configuration through generative computational systems.

The discussion of motivations in chapter one described the shortcomings of academic and professional R&D into computational design as perceived by the author around the year 2000. Concerns of design computation were not compatible with the fundamental architectural philosophy of human-centric spatial design espoused by architects and theorists and the discipline of design computation seemed stuck in explicit quantitative optimization or aesthetic form-finding (Derix 2014). Three strands of academic research cornered specific areas that each appeared to explore one aspect of a computational system for human-centric space design. Those were discussed in chapter three and were grouped around (a) the mathematical representations of environments by some members of Cambridge's LUBFS, (b) syntactical representations of spatial configurations by Space Syntax and (c) distributed generative algorithms for designing architectural spaces by the New Epistemologists.

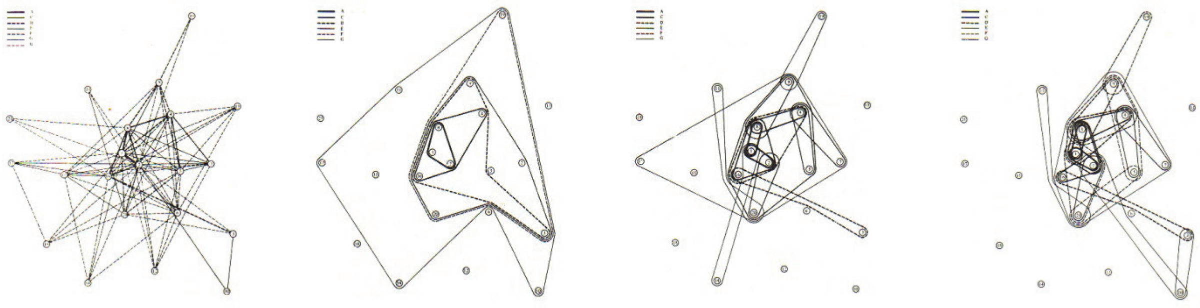


***Fig321**. Philip Tabor (1971): k-dendrograms showing the spatial clustering of locations into operational overlaps by walking distances; Tabor called this representation the 'associative web'*

Members of LUBFS with Philip Steadman as the main exponent developed many fundamental mathematical representations for architectural computation and spatial analysis as discussed in chapter 3.6. Mathematical and systemic representations of architectural form and spatial operations were explored including occupation (Willoughby 1971; Tabor 1971). The focus of computational representation shifted from space to process in the 1980s with the advent of the Knowledge-based Design discipline, which attempted to formalize all procedural aspects of the workflow based on decisions, which in turn rest on externalized learned knowledge (Gero 1990).

Design heuristics became a focus for explicit automation into computational heuristics (Akin et al. 1992) that were not generalized into meta-heuristics. The developments at LUBFS and the KbD discipline therefore provided one aspects of human-centric computation for spatial design by introducing mathematical representations of space and conceptual thinking of encoding design heuristics into workflows.

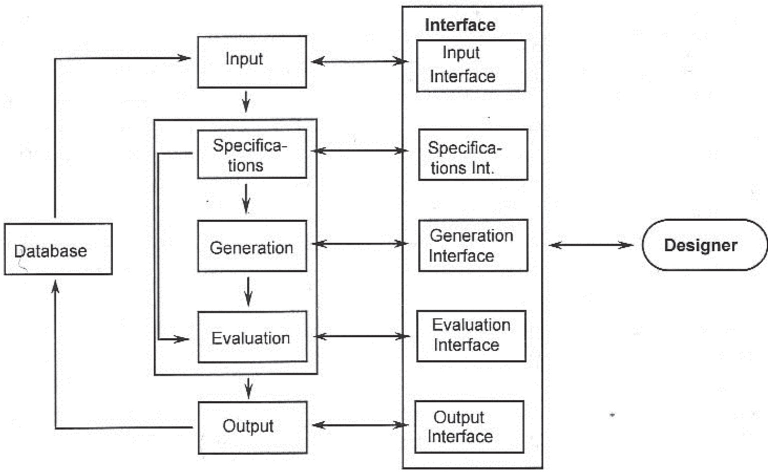


***Fig322**. Ulrich Flemming (1994): the SEED layout planning system based on a database of designed cases represents a typical 'knowledge-based design' automation approach*

The main contribution of Hillier and colleagues' Space Syntax to architectural computing rests on the development of methods for the analysis of spatial configurations of cities and buildings that are partially derived from human agencies such as movement or visibility. While Hillier revealed correlations between spatial structures and human-centric affordances (Hillier 1996), it was mainly Alasdair Turner who developed and generalized syntaxes into computable algorithms (Turner 2000; Turner et al. 2001). Analytical syntaxes for spatial configurations based on occupant-centric agencies provide the second aspect of human-centric computation for spatial design.

New Epistemologists are mainly represented by Paul Coates and John Frazer. Coates and Frazer developed the first *bottom-up* distributed algorithms for generative design in architecture[94], using concepts from cybernetics, systems theory, artificial intelligence and life (Frazer 1995; Coates 2010). Their concern with autonomous self-organizing systems provided architectural computation with meta-heuristics that encode dynamic generative methods. Meta-heuristics simulate generic strategies approximating consensual system states rather than precise procedures for optimization. Hence they represent open systems that can be applied to a large range of complex spatial design contexts. Self-organizing algorithms with distributed representation for generative systems provide the third aspect of human-centric computation for spatial design.

[94] As discussed in chapter two, contemporary developments exist such as Georg Stiny's shape grammars that were however more concerned with replicating existing patterns rather than generating new knowledge.

Little integration existed between those strands until around 2005[95] and academics like Paul Richens (1994) and Robin Liggett (2000) had highlighted the problem with implementation in practice while the architectural profession was focussing on CAD software. The only isolated attempt at a concerted effort to integrate generative with configurational syntaxes into a human-centric planning system dates back to Hillier and colleagues' original Space Syntax generative theory of urban settlements (Hillier et al 1976), which was encoded into computable syntax by Paul Coates (Hillier and Hanson 1984). Therefore, throughout this dissertation, Hillier's early research into spatial planning systems and Coates' developments of design algorithms have been used as two core references.

With the founding of CDR in 2004, the author began to outline the balancing of the strands in order to focus on the design of user-centric spatial configurations. Chapters five to eight provide case studies that illustrate the generalization of the foundational academic research through pilot application to design projects with real-life constraints from urban planning and architecture. The order of those chapters crudely reflects the necessary logical stages in the development of a USOM.

Chapter Five applied generative algorithms used by Coates and Frazer to briefs of urban and building design with a focus on spatial issues such as spatial planning (urban structures via road networks, accessibility and block definitions) and space planning (adjacency diagrams, layout design, massing compositions and interior design). The chapter shows how generative meta-heuristic algorithms such as evolutionary algorithms (single and multi-criteria GAs), agent-based systems or cellular automata previous explored academically are relevant beyond a theoretical discourse in architecture and can eventually lead to real-life results[96]. This required the elaboration of data structures to be less abstract and evaluation functions to become more stable. But most importantly to identify appropriate design processes from professional practice whose heuristics logically correlate to meta-heuristics and objectives align with the algorithmic epistemology. New distributed self-organizing algorithms were explored not covered by the New Epistemologists like the Pareto multi-criteria optimization, force-directed graph grammars or simulated annealing. Through iterative implementations, the meta-heuristic models became sufficiently generalized to be readily applicable to a range of scales.

Chapter six explored spatial analysis models of Hillier and colleagues under varying design conditions. CDR have extended Space Syntax's repertoire in many ways: from two to three and four (time-based) dimensions, from post-design evaluation to real-time interactive *designing* application; and through new representations of configuration and human-centric affordances such as medial axes networks and self-organizing neural networks. All models were generalized through iterative implementation and many models supported real-life projects that were realized such as the National September 11th Memorial Museum in New York or the Packages

[95] As discussed in chapter one: programming API's became more accessible and the first students of UEL CECA and ETH CAAD started to work in practice.

[96] Projects realized or under construction from this chapter include: ENK Complex, VITA and many layout planning stages of real projects from the Layout Tool (see 5.2)

Retail centre in Lahore. The development of self-organizing artificial neural networks in the design process constituted a novelty for the field of architecture and for its limited application represents a less robust generalization to date. Also the use of medial axes for the syntactical generation of spatial structures and its application to the configurational analysis in a live-project setting constitutes a novelty in the architectural profession (as well as academia where it has been only cursorily looked into by March and Steadman (1971), Michael Batty (Rana and Batty 2004) for urban structures or Wiener and Franz (2008) for building navigation).

Chapter seven demonstrated the mediation of generative algorithms with human-centric spatial analysis into design processes. The two strands are first integrated traditionally by evaluating generated spatial configurations by human-centric KPIs or configurational objectives, akin to the standard *generator-test-cycle* (Simon 1968). However, the application of professional KPIs represented as fitness-functions for human-centric or configurational performances as shown in chapter 7.1 was new. The process is extended by situating the designer through interaction into the algorithmic model. This in turn requires the model to be less procedural (such as a GA built on generations), the structure of the design process to be hierarchically flatter and explicit target states removed in favour of the designer functioning as interfering observer and simulated occupant. The Euston Pilot in chapter 7.2, demonstrated how learned spatial intuition and design heuristics are employed to mediate the model. Finally, ANN models for associative planning of section 7.3 introduced a completely new design process by evolving the design schemata in parallel to design states. This cognitive correlation between spatial states and perceptual qualities has not been tested in depth yet and has not allocated a clear role for the designer as yet. Its basic algorithms however have been tested and generalized for stable use and add value to strategic design processes through spatial classifications (such as Packages and many current projects). Mediated design processes of this chapter extend the knowledge-based models such as Akin's HeGel or Gero's 'function-behaviour-structure' model who sought to automate heuristic and cognitive design processes. They are more successful by integrating the designer as acting observer into the mediated workflow stimulating unpredictable empirical and cognitive capacities rather than simulating known creative design heuristics (Akin 1998).

Chapter eight discussed the development of design systems for designing, called meta-systems. It combines all research strands discussed in chapter three into an Open Framework for Spatial Simulation (OFSS). This final stage in the definition of a USOM loosely associates a series of heterogeneous algorithms and representations for spatial generation and analysis into a single computational design framework. This associative structure abandons the standard hardcoded dependencies of parametric representations and removes the bias of workflows towards intentions of the designer-observer, algorithmic epistemology or the analytical constraints. A spatial meta-system has been established that does not tautologically pre-empt its own phenotypes due to hierarchically fixed ontologies. It realizes Hillier and Leaman's *manifold* (1974) and completes Stan Allen's (1997) forecast that field conditions are not only for graphic patterns but also user behaviours. Academic case studies demonstrated how theoretical concepts of Hillier's and Coates' syntactical planning system were tested to understand their associative structure for professional workflows and were generalized for live projects that are being realized (EXPO 17 Green Block).

9.1.1 Contributions to Discipline

The first contribution of the first research objective represents an *epistemological shift*: what can be designed through computational algorithms in architecture and urban design? The focus of what can be known from computation is slowly shifting from quantitative optimization of explicit performance targets to qualitative configurations of implicit performance behaviours. Associative models in chapters seven and eight represent some of the first pilots in the field for this shift towards a 'new organic' *inside-out* human-centric spatial computation (Derix and Izaki 2013) and have been widely discussed in academia and industry. The basis for achieving this turn of computational design focus lies in the translation, association and generalization of the three discussed strands into *designing* processes to demonstrate the feasibility of this shift. Coates anticipated this new model of *knowing architecture* through autonomous algorithmic representations, calling it the New Epistemology (Derix and Izaki 2014).

Secondly, in order to arrive at working associative models, new algorithms from other disciplines and novel representations had to be introduced that were hitherto untested in architectural computation and were discussed in chapters five and six. To enable distributed self-organizing meta-heuristics to be used outside isolated academic research, new algorithms such as the quantum annealing and Pareto optimization had to be introduced that extend traditional evolutionary algorithms. Similarly, instead of using the orthodox cellular automata, recursive graph traversal algorithms were introduced. To explore cognitive qualities of space, Kohonen's SOMs and Fritzke's GNG were introduced to the field, which are currently leading to a novel spatial classification methodology (Derix and Jagannath 2014a; 2014b). New representations for existing spatial analysis algorithms such as Benedikt's isovist, Turner's visual graph analysis, Dijkstra's paths or Blum's medial axis had to be developed to translate their application from urban plans in two dimensions into building volumes in three dimensions.

9.2 OBJECTIVE 2

Develop a structure that best facilitates a computational system for human-centric generative design.

The second generation of KbD has paved the way for very large software packages such as BIM attempting to realize Christopher Alexander's (1964) initial aim to categorize all aspects of the design process into sub-categories that 'simply' require pre-processed organization into target state-based solution paths. All-integrating models were called *monolithic* as they attempt to incorporate all ontological and epistemological design domains hard-coding their exhaustive associations. Similar thinking often permeates academia and early models of the author such as Faulty Towers (5.1) or SynUrb (8.1) illustrated this approach when initially moving from university to practice. Over a series of projects in practice the lessons of Rittel and Webber were learned and it became clear that a computational design system aiming to simulate human-centric affordances or simply to work within a live project context cannot pre-empt solution states and reverse engineer ideal predetermined solution paths (Rittel and Webber 1973). The specific structural transition to decompose monolithic models into a semi-automatic process of autonomous components was described in 8.3. However, structural organization of the computational system hinges also on other representational and functional developments that were partially discussed within the case studies and are summarized here:

9.2.1 Meta-Heuristics

Heuristics are an individual's rules-of-thumb to approximate a specific state, be that through an analogue design procedure or a computational algorithm procedure. Most computer programs in design settings are developed as quick tools with heuristics permeating the software. It is very difficult to transport or scale heuristics. For live *designing* generalized routines called meta-heuristics are required that can quickly be adopted and applied within a project's lifetime. Heuristics are produced during an instantiation for a project to serve as interfaces between meta-heuristics. Computer programs based on individual heuristics or project-specific constraints are not usually re-applied (Liggett 2000; Miranda and Derix 2009).

The generalization into a meta-heuristic leads to the culling of bespoke functionalities and data structures reducing models to parsimonious representations (Coates and Derix 2007). Parsimonious models produce smaller, more concise components with agile adaptation potentials (Miranda and Derix 2009). Concise components enable easier accessibility by reducing the I/O endpoints into a reduced number of general data formats such as DXF specifications.

9.2.2 State Search

Fig323. *Frei Otto, 1960s: catenary and tensile models represent 'live' material computation whose behaviors are autonomous from the user who has to interact to learn their dynamics (Otto and Rasch 1995)*

Deliverables for professional design processes are mainly expressed via quantitative targets, measuring compliances from regulation. Even without computational design, human-centric design possesses few explicit performance measures as highlighted throughout the dissertation. Many computational models for the configuration of human-centric environments generate complex distributed states rather than discrete solutions to meet explicit targets. Meta-heuristic search algorithms are better constrained via design objectives than explicit targets. As discussed in chapter 5.1, it is difficult for an observer to simply impose targets when a distributed system produces states by consensual performances. Distributed performances represent behaviours which the observer cannot directly affect. The observer needs to learn to respond and propose in a system of meta-heuristics in order to generate a system behaviour that produces desired results. Otto's catenary models were exemplary where the designer interactively explored behavioural states of an autonomous physical system in order to mediate his intentions into it (Otto and Rasch 1995). Hence, it is desirable to limit the complexity of individual simulation models to be able to mediate meta-heuristics and affordance performances towards a satisfactory state (Derix 2010).

9.2.3 Role of the Observer

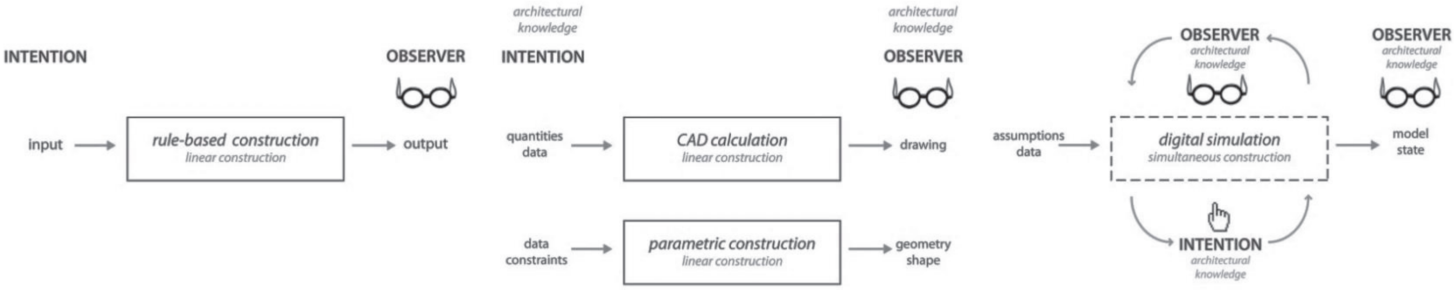


***Fig324**. Role of Observer: the role of the observer hardly changed from an analogue (left) to a CAD and parametric design process (middle); in an algorithmic simulation with visualized interactive behaviours such as Otto's physical models, the observer enters into 'structural coupling' with the model's heuristics*

The role of the observer as discussed throughout the dissertation is changing from author to mediator, and also to learner. The romantic image of the New Epistemologists desired a completely autonomous authorless design system (see 2.4); yet such a system does not allow for easy integration into heterogeneous spatial design settings. The observer who is addressed in this dissertation is the designer who represents several user agencies: (a) as observer of the computational behaviours, (b) as observer of occupant perceptions and behaviours that he needs to simulate mentally (empathy) and syntactically, (c) as designer who observes and mediates algorithmic processes and (d) as communicator to other stakeholders. While this carousel of roles applies mainly to the new computational designer, it also partially applies to traditional architects and clients who make increasing use of software. In order to facilitate fluid switching of roles and observing/facilitating agencies, no specific place should be allocated to the designer within this meta-system, leaving the framework as permeable as possible.

9.2.4 Visualization Immediacy & Interaction

An emphasis in the case studies was given to visualizing behaviours by attempting to render state changes of spatial units promptly (for example chapter 7.1). Meta-heuristics and the analysis of effects of spatial configurations on occupants implicitly encode behaviours either by simulating *designing* heuristics or cognitive decisions. Hence, it is vital to reveal those behaviours and correlations to facilitate empathic responses from observers of all types. Philosophers Barnes and Thagard (1997) of the Waterloo Computational Epistemology Lab describe the relationship between simulation and the observer as a case of empathy by simulating someone or something else's intentions by creating an internal analogy unconsciously, meaning without knowing the opposite's reasoning (Barnes and Thagard 1997). In other words, the observer does not have to understand the syntax of algorithms but by being able to visually following decisions and actions undertaken by a model, he can identify with the generative path by understanding where his intentions align or deviate from the process (Derix et al. 2010). He can tacitly compare his heuristics with the meta-heuristics, creating a consensual narrative domain. Consensual domains are built on *structural coupling* (Maturana 1978), which like Vischer's aesthetics and Lipps' empathy rely on isomorphic alignment of physical and mental structures (Schwarzer 1991). Reminiscent of Otto's physical model set-up, interaction vastly improves this empathic process because the observer can verify his evolving understanding by watching his interferences absorbed into new states. Subject to resulting states, he can check whether the model behaves according to his learned intuitive expectations. Intuition debugs the algorithmic model.

To visualize dynamics as clearly as possible, models should represent as few behaviours as necessary, processing a small number of drivers that inform configurational states. When too many behaviours process simultaneously into a complex configuration, it becomes increasingly difficult to visualize those behaviours, associations and their spatial correlations. A *black box* effect entails limiting intuitive access to the learning process of the system. Observers not familiar with meta-heuristic simulation will revert back to known deliverables and targets because only the final quantifiable states would be assessable instead of intuitive performances (Derix 2010; Derix et al. 2012). As discussed in chapter 8.3, stakeholders cannot identify the correlation between decisions and process of a *black box* model.

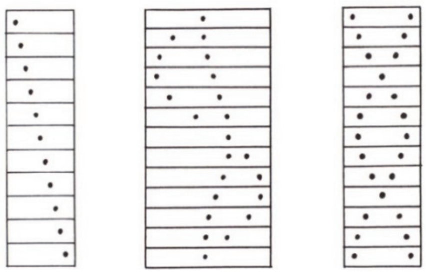


Fig325. *Rudolf Arnheim (1974): stroboscopic movement, showing time-based state changes perceived as behaviour; image composed by author from originals*

The generalization of heuristics and workflows into meta-heuristic representations helps the visualization effort. A parsimoniously lean model contains generally fewer behaviours and representations whose dynamics are easier to visualize through step-wise rendering updates without greatly slowing the simulation. Rudolf Arnheim (1974, p372 – 409) summarizes aspects of perceiving phenomena such as movement of objects and behaviours. Employing Gestalt Psychology principles, he repeatedly identifies structural simplicity and state-change sequences as two vital aspects for the observer to visually perceive dynamic phenomena, instilling a sense of purpose.

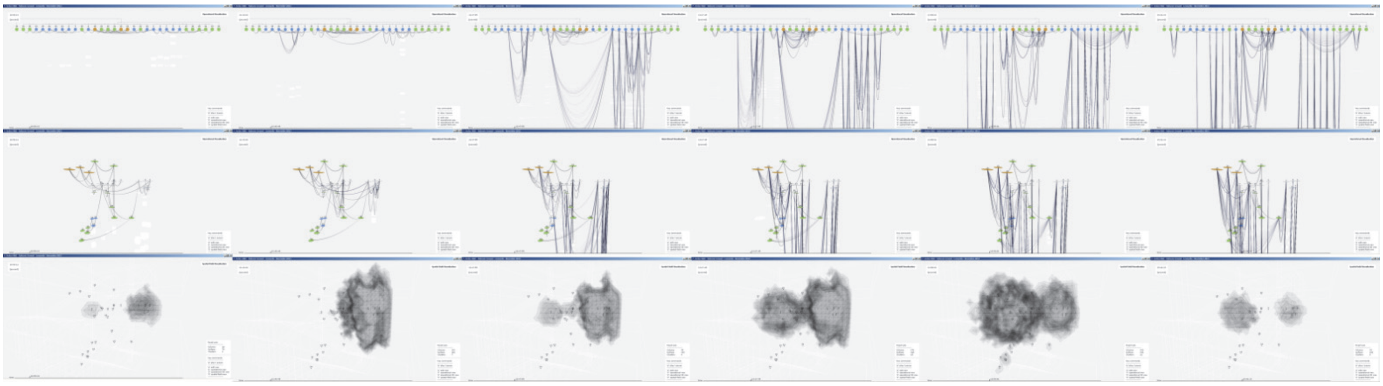


***Fig326**. Near-Living Architecture (NLA), 2013: an example of a lean behavioural model with clear state-change sequence for diverse aspects of the design simulation: three interfaces for interaction and visualization of behavioural, operational and spatial states for the NLA project with Philip Beesley (Izaki and Derix 2014): six snapshots of the (top) topological activation; (middle) spatial operation and (bottom) spatial clusters*

9.2.5 Contribution to Discipline

The contribution responding to the second research objective represents an *ontological shift*: what is the structure of a computational design system that enables *designing* for human-centric spatial configurations through computation? It is neither an epistemologically closed system as modelled by KbD, nor an epistemologically authorless system of the New Epistemologists or purely constraint-driven evidence structures of Space Syntax. Instead, it is represented by an open meta-system of associated representations for spatial, behavioural and cognitive affordances, synthesizing all three R&D strands representations into a semi-automatic observer-situated system. A field of actors in which the observer exerts heterogeneous agencies.

The OFSS is a first prototype of a meta-system for design. Its structure inherently facilitates instances of human-centric spatial configurations, associating spatial, meta-heuristic and analytical components. This structure is based on a range of structural qualities:

- Parsimony generalization of heuristics into meta-heuristics
- Openness mediated semi-automatic workflows
- Interaction situating the observer to mediate the state search
- Visualization immediacy of rendered behaviours for empathic identification
- Agency heterogeneous agencies for observer, not fixing his role

The ontological structure occupies the middle ground between the extremes of Alexander’s and the KbD fully automated observer knowledge, and Paul Richens’ and the profession’s fully fragmented observer-controlled CAD system. The meta-system allows for diverse spatial schemata to emerge through Donald Schön’s (1983, p185) *generative conversations* in computational design illustrated by FuCon or Floating Room and enables speculative theories such as Humberto Maturana’s *consensual domains* via cognitive structural coupling as tested on RIBS (Maturana 1997). Unlike apparently open professional platforms like McNeel's Rhino[97], the meta-system is structurally closed because its spatial ontologies, algorithmic behaviours and cognitive affordances are generalized into an associative structure, yet informationally open since interfaces to any external process via general data formats and mediation of observers are enabled. This ontological representation complies with the fundamental concepts of open complex systems (Cilliers 1998).

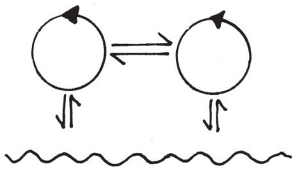


Fig327. *Humberto Maturana (1978): diagram of structural coupling between autonomous systems, being structurally closed yet informationally open within their environments (circles representing systems, arrows represent perturbances and the wavy ground line the systemic context)*

9.3 LIMITATIONS

While the OFSS represents a state-of-the-art computational system for a specific design methodology, there are shortcomings and open questions. Some of its structural benefits also contain weaknesses. So far, in order to guarantee a space-human correlation, other aspects of the general design workflow have been isolated such as scaling-into more detailed issues in later stages of schematic design stages. Besides the focus on the correlation, an up-streaming could be detected, meaning that the OFSS is used in ever more abstract contexts of strategic design. The draw towards earlier decision-making pre-design stages must not necessarily be seen as a limitation but it does not allow for a thorough validation of the early design simulation results.

Validation of generated data and configurational outputs represent another weakness. While the theories, algorithms and representations used are well researched in academia, some of the novel developments go beyond university knowledge. Clients can help to verify results and close collaboration with public and private bodies ensure constant monitoring of developments, yet distinct verification phases are not regular and should eventually find their way into the workflow.

The associative structure of the OFSS is *structurally closed*, so that interfacing with other software is not difficult yet only 'informationally open'. The framework is programmed in a single programming environment. It can interface with libraries using other languages and through bespoke protocols as demonstrated on FuCon. But it does not provide a very large spectrum of I/O formats as other CAD packages do.

[97] McNeel Rhino is a 3D modelling software with many plug-ins and programming interfaces. But it is neither focussed on human-centric and spatial design, nor networked and associated into domains like the OFSS: http://www.en.na.mcneel.com, accessed 09.01.2015

It is not a declared aim to make it *open source*. OFSS is developed and maintained by members of SUPERSPACE and the professional setting prohibits a completely open source approach, which might or might not be beneficial to such R&D.

The single stand-alone programming IDE without open source approach requires longer training and practice of team members in computer programming. There are occasionally affiliated colleagues who know some *scripting* but find it too difficult to slot into the framework for design development. It appears a high hurdle to join in from the outside.

9.4 EFFECTS ON PRACTICE

The use of computational design has become main-stream for larger practices with regards to parametric modelling for complex geometries such as facades, physical structures or occasionally some formal aspects of form-finding (plus climatic analysis and project management). But aspects of space and spatial planning, let alone human-centric space design are still absent, bar a few exceptions[98]. Hence, the impact of human-centric space design is still relatively unknown. But a few effects observed so far from the use of the OFSS will briefly be touched on.

9.4.1 Organization of Design Process

As described in the *Role of the Observer* above, depending on the brief for a development of an instance of the OFSS, the user can be any stakeholder of the spatial environment process. Yet, the user of the OFSS itself will always be a member of the SUPERSPACE group. When commissioned directly as a proprietary software for a third party client, instances are compiled into executable applications for the client's internal use such as the Near Living Architecture project for Philip Beesley architects (Izaki and Derix 2014) or the web-based server-client interface of the VITA shelving project. But the vast majority of instances are used by the group itself, serviced in design settings.

GENERALIZATION

Instances of OFSS are either developed during a live design process or a funded research project. The most difficult but common scenario is development during a live design process where alignment with a standard architectural workflow is necessary. Having generalized many of the most common representations and algorithms allows for fast adaptation. Generalization often happens through re-use or iterative requests for similar design applications. Repeated applications are not necessarily all spatially oriented but include many interfacing and functional elements such as common interaction functionalities like 'picking' and 'dragging' 3D points. When spatial aspects and behaviours are recurrent, a meta-heuristic can emerge, because idiosyncrasies of individual designers are culled to fit a standardized heuristic that applies to similar design contexts. This is for example the case for topological or bubble diagrams (see chapter 5.2 *Parts Assemblies*).

[98] Even where groups like CDR exist within commercial practice, it is only employed on a very small number of spatial planning projects at any one time.

Generalization takes place over at least two stages: a specific heuristic or spatial aspect conducted during a live project is (1) generalized directly after the project or (2) in follow-up projects and funded research. The first stage is mostly complemented with a second stage that is often conducted by another computational designer with a different approach. Because the case studies were not discussed chronologically but thematically, this development process was not explicitly visible. For example, the first bubble diagram was produced by the author in 2005 using multi-agent systems, the second by Pablo Miranda in 2006 using graph grammars, the third by Lucy Helme from 2009-13 using multi-agent systems with physical forces (Helme et al. 2014) and a current iteration conducted by Åsmund Izaki builds on a blend of previous algorithmic representations and behaviours, using multiple generalized OFSS components.

Occasionally and subject to resourcing, this iterative shared development process can occur within a live project design. On FuCon and SSSP (see 8.2), simultaneous development made cross-overs between design codes possible and therefore increased efficiency both in terms of speed and depth (the Agile methodology). Generalization then happens during the project lifecycle, not iterated across projects.

WORKFLOW

The iterative development process – fast or over time –of small chunks of structural and behavioural code into generalized components is reminiscent of the Agile manifesto (Miranda and Derix 2009). The Agile approach is based on good practice of software development[99] with fast cycles of parallel developments of similar design aspects and many client workshops for feedback (Highsmith 2000). While cycles in a live project setting are indeed fast and meetings with the collaborating architecture team continuous to align objectives and deliverable formats, parallel development rarely takes place. Generalization over many projects on the other hand by different computational designers scales up the Agile approach into parallel development and relatively small cycles (weeks and days instead of days or hours).

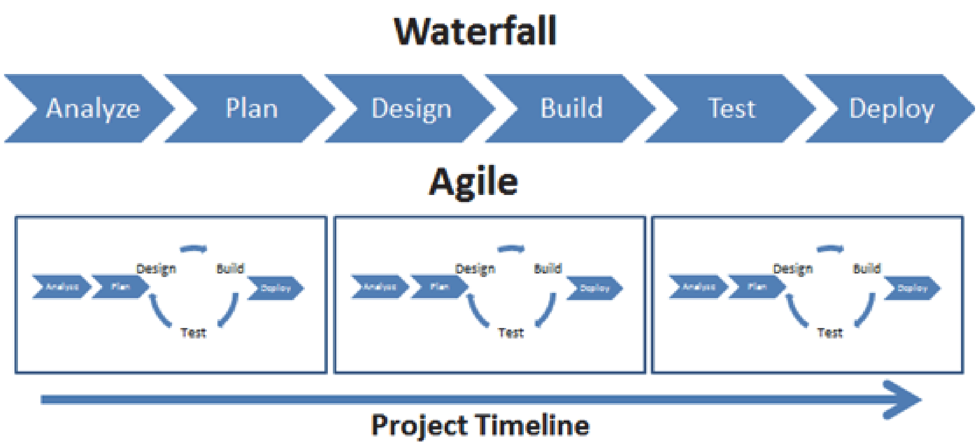
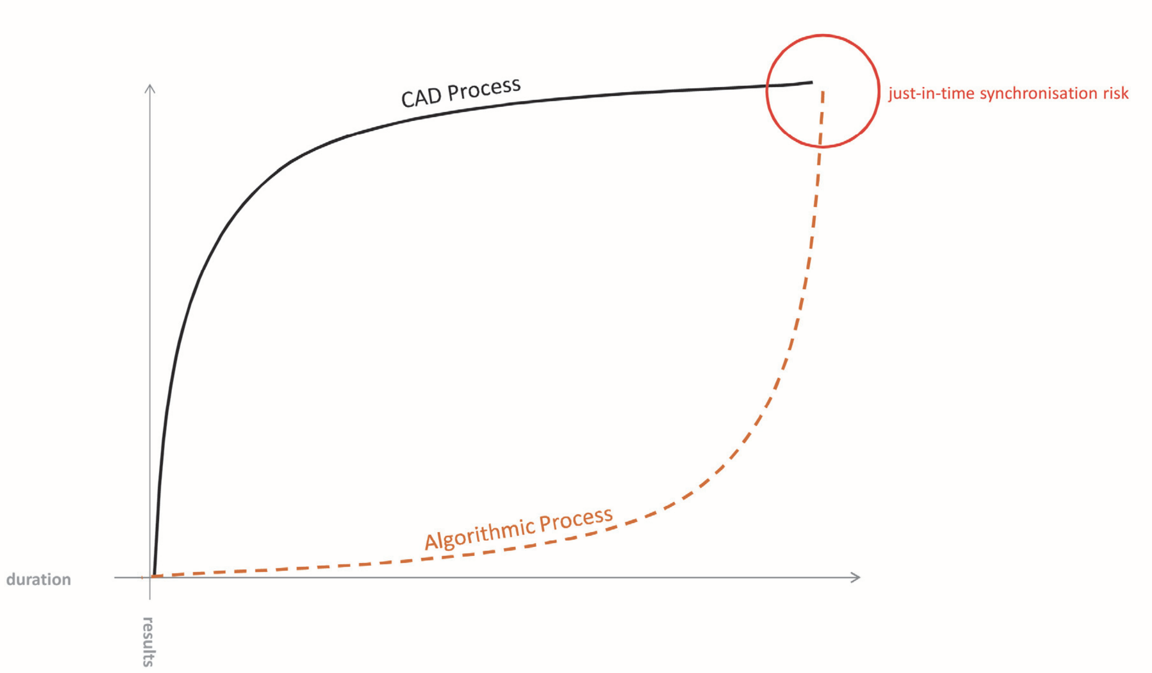


***Fig328**. Agile vs Waterfall development (Highsmith et al. 2000): the waterfall process is reminiscent of the Simons' target-state solution path, while the Agile process represents a quick iterative prototyping method*[100]

[99] Online manifesto: http://agilemanifesto.org, accessed 09.01.2015
[100] Image from GreenLine Systems: http://www.greenlinesystems.com/agile-software-development/, accessed 09.01.2015

Clearly, big differences exist between developments in architecture and software: architectural computing does not represent a public-facing commodity but a bespoke project-based workflow between few people. And where in software development a loose brief is set by a client with developers evolving the brief, in architecture the clients' brief is set *a priori* and remains relatively stable. In architectural computing, two interpretations of the brief emerge simultaneously from the collaborating computational and architecture teams. The timescale of developing software rather than drawings and specifications are structurally inverse proportional: code takes longer to converge but enables fast adaptation later, while conceptual drawings are done fast and take longer to converge towards schematic interpretation. Hence, even if quick cycles of development proposed by Agile were envisaged, development stages are hard to align along the same milestones between teams with different representations.



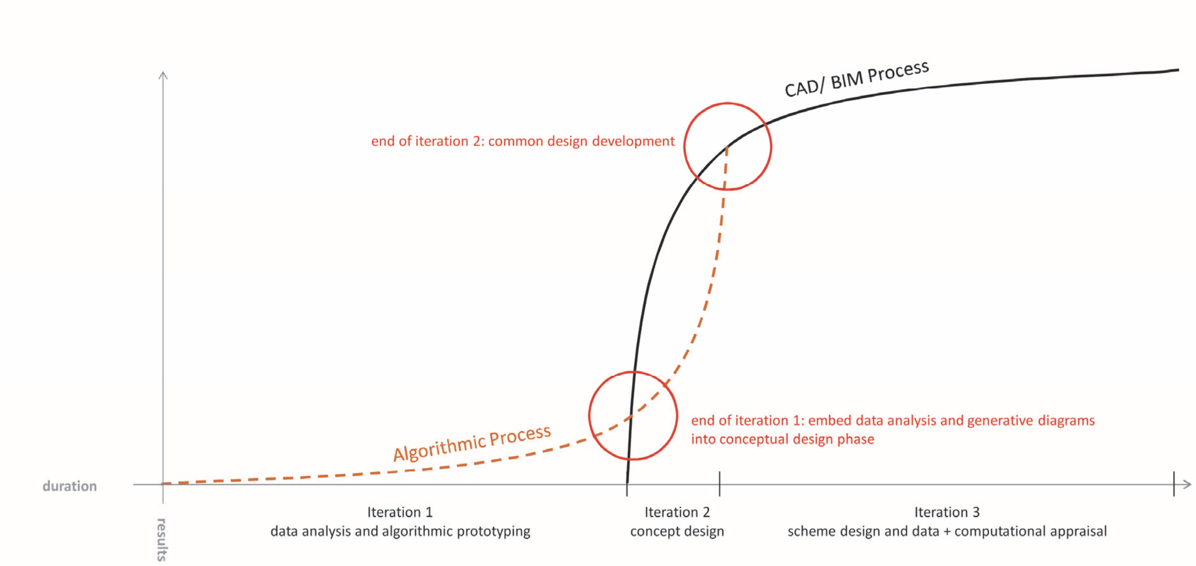


***Fig329**. Algorithmic vs CAD design development process: (top) as is and (bottom) adjusted to integrate better*

Hence, three types of development workflows for computational design result, depending on speed and matching representations: (1) short collaborative projects like competitions with high usage of generalized code components of OFSS, slotting into an overall design workflow, akin to the mixed-media *Floating Room* case study; (2) medium-long design such as conceptual or schematic design phases with a mix of new code and generalized components. Here generalized applications such as *Routing* are often combined with new applications into mixed workflows; (3) long design duration for mostly new developments such as the *National September 11th Memorial Museum*, conducted outside standard workflows.

Fig330. *Pattern Language, Christopher Alexander (1977): a catalogue of spatial properties as graphic patterns that are theoretically combinable into designs without rules*[101]

It is mostly the third workflow that leads to generalization of structure and behaviour, then tested again in workflows of type one. This generation and utilization of general code components is more closely related to what the software development industry calls Design Patterns. Conceptually based on Christopher Alexander's Pattern Language (Alexander et al. 1977), Design Patterns are functional *snippets* of object-oriented code that serve as structural and behavioural skeletons for development of bespoke software (Gamma et al. 1995). The OFSS provides however also an ontological and epistemological domain towards spatial human-centric design computation and thus represents an new hybrid between bespoke software and Design Patterns, more akin to a spatial Software Development Kit (SDK).

[101] Image retrieved and adjusted from: www.tkwa.com, accessed 10.01.2015

Two other aspects regarding workflow integration are not elaborated: instead of stand-alone applications, models can be delivered as separate components via plug-ins to CAD packages (done from 2004-7 but abandoned). And the balance between interaction and automation, i.e. the 'human-in-the-loop' relates to the discussion of parsimony, visualization and semi-automation.

9.4.2 R&D Life-Cycle | Infectious Epistemology

There are broadly speaking two relevant development cycles: behaviour-to-procedure (b2p) and procedure-to-behaviour (p2b). b2p means heuristics from specific contexts like sectors and stages abstracted into generic meta-heuristics. And p2b are meta-heuristics that inform contextual behaviours (Derix and Gamlesaeter 2012). As in the SSSP project, correlations between design heuristics and algorithms were sought and modelled through various algorithmic prototypes. Those were generalized over time to become repeatedly serviceable meta-heuristic components. In this case, analogue design informed algorithmic development. Reversely, qualities of meta-heuristic components can be integrated into standard design methodology, producing new design behaviours (p2b). In this case the algorithmic model informs the analogue methodology.

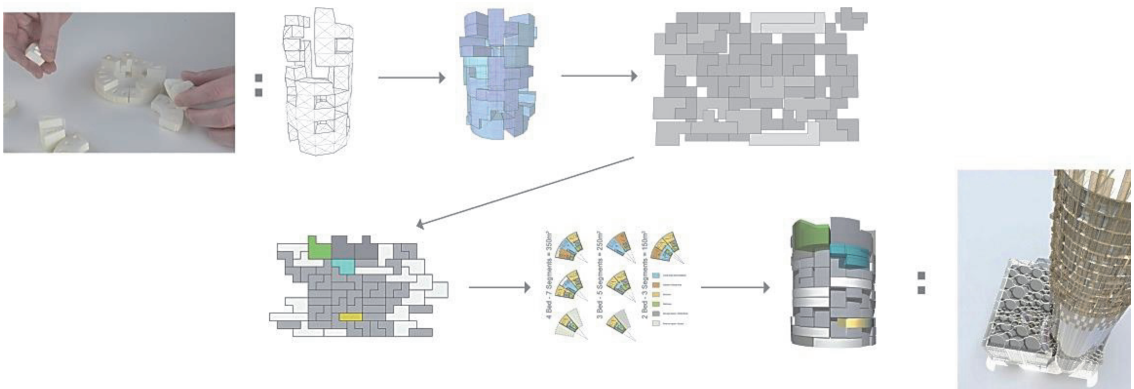


Fig331. *b2p on Khalifa-bin-Zayed competition, 2009: analogue design heuristics are encoded and evolved into a meta-heuristic during a competition, utilizing three types of EA; the meta-heuristic is later generalized for the OFSS*

A hybrid between the two development cycles occurs when a generalized algorithmic model is not only stipulating new design methodology (p2b) but also leads to new analogies in different design domains. For example, the above mentioned adjacency diagrams were not only iteratively elaborated over ten years but also gave rise to parallel developments with different representations. Some building capacity models based on evolutionary algorithms where building envelops have been designed *a priori*, resulted from the adjacency and layout models using different representations. Hence, a parallel b2p was initiated for a similar problem with different heuristics and constraints such as transport station box design (usually underground) or building re-use. As a consequence, a new workflow is slowly emerging using evolutionary packing models. B2p (bubble diagrams) gave rise to p2b (layout design) sparking a parallel cycle of b2p (capacity) and p2b (refurbishment).

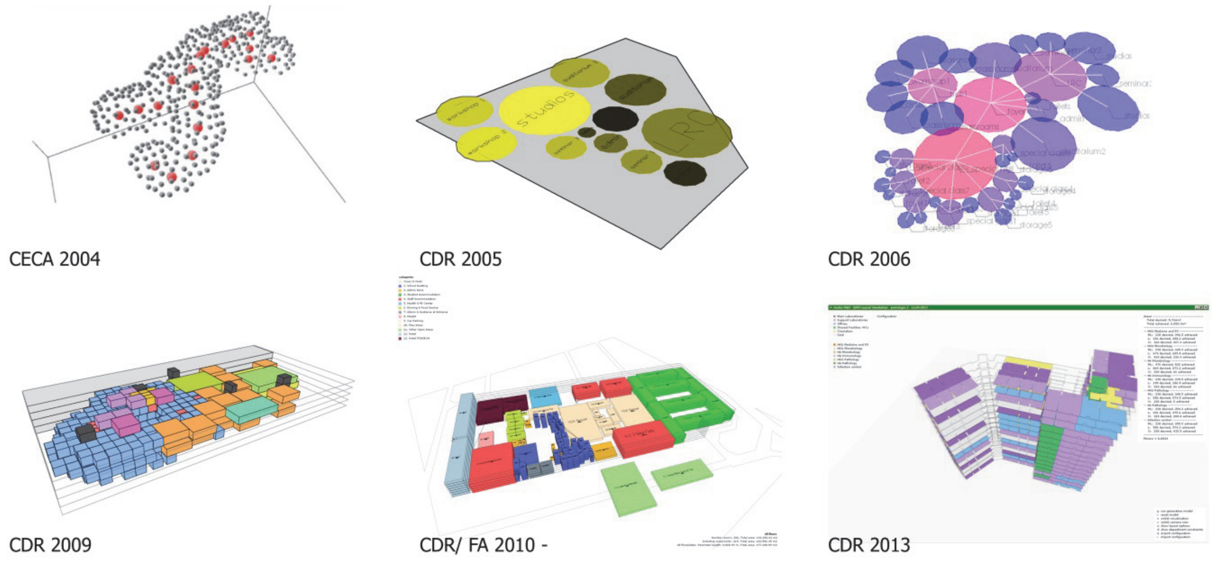


***Fig332**. p2b for layout planning: from observed natural behaviours at CECA 2004 (top left) to design heuristics at Aedas 2005 (top middle) to graph generalization 2006; this spawned another strand of previous agent-based meta-heuristics in row below: 2009 ADEC competition heuristic, 2010 meta-heuristic for general layout and 2013 building refurbishment as a new strand of applying a meta-heuristic, changing the traditional approach*

The false split into *routine* and *creative* design proposed by KbD becomes obvious. Intuitive and creative heuristics gives rise to algorithmic routines with their inherent epistemic bases, containing elements of related design knowledge domains. Underlying domains of OFSS refer to space-user correlations that when abstracted are self-similar across many typologies and scales. One epistemic model infects others and recycles into new domains.

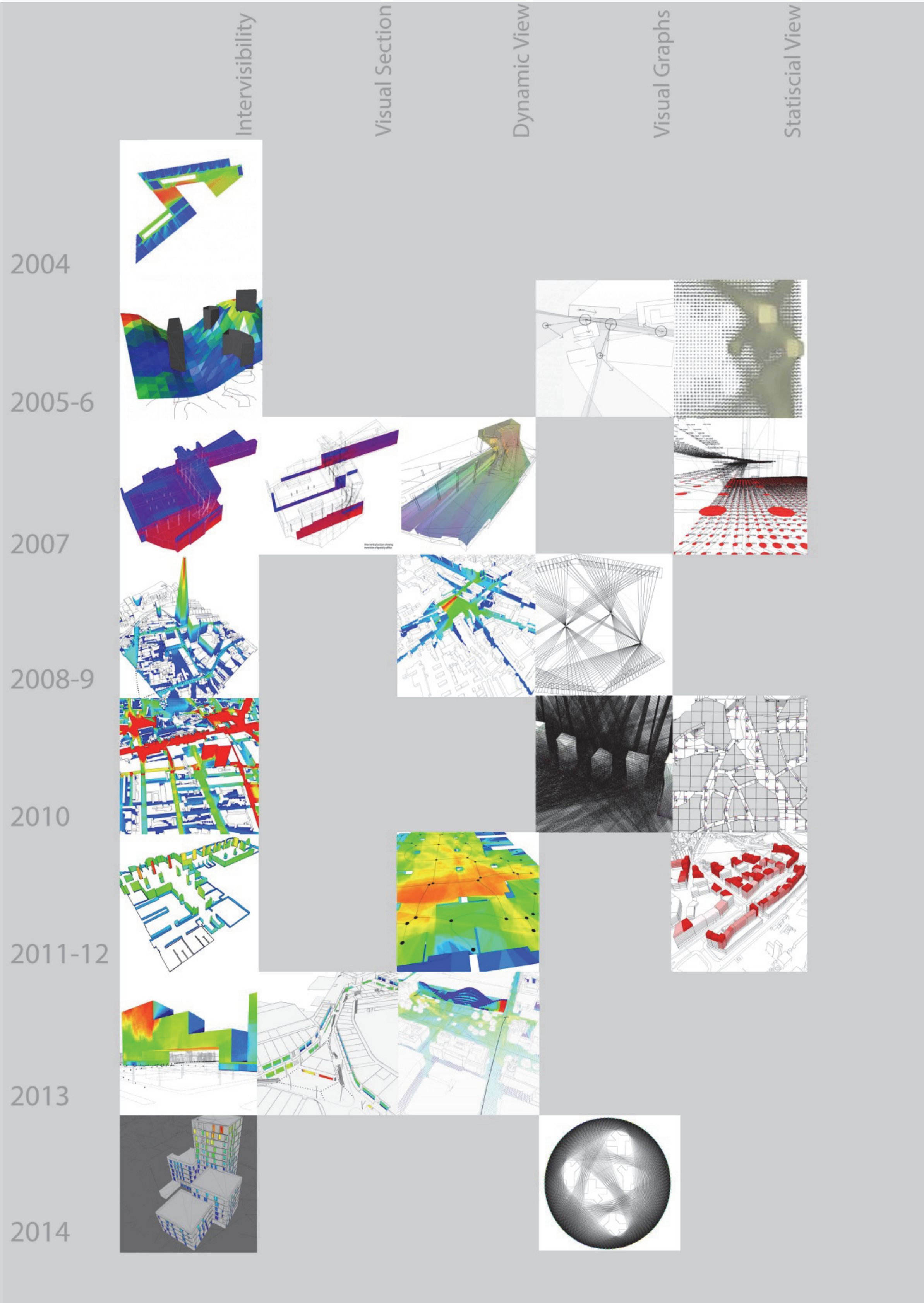


Fig333. *b2p and p2b: for ten years, visibility analysis models have been developed that spawned new behaviours and procedures; most have been generalized into the OFSS and become integrated into the practice workflow*

APPLICABILITY

"A system which meets commercial needs of today should provide interface capabilities ranging from complete user interaction, where the user interactively specifies the location of each activity, to complete automation, where an algorithm generates an initial solution. Or as desired, a designer should be able to interactively locate some activities and use an algorithm to locate or suggest locations for others. Rather than generating a single least-costly plan, the designer with the aid of automated algorithms can make trade-offs between competing criteria and converge on a solution that responds to a broad spectrum." (Liggett 2000, p212) Robin Liggett provided some insights for why academic research into layout planning is not being applied in practice. Apart from interaction and interface aspects, the key observation relates to above described *monolithic* academic research model where single cost functions and automation do not align with professional *designing* methodology.

As observed by Liggett, the industry has taken a different path from academic research. Practice-based R&D groups have managed to build trade-off systems for routine design that appears inappropriate for academia, concentrating mainly on creative design[102]. From the author's observations at universities, this has many reasons amongst them: insufficient teaching of the actual (writing) skill of algorithms in architectural design; staff lacking dual knowledge of theoretical and practical application; master and PhD students are limited to 2-5 years, usually working in isolation; therefore, leading to a lack of real-life constraints and pressures to explore.

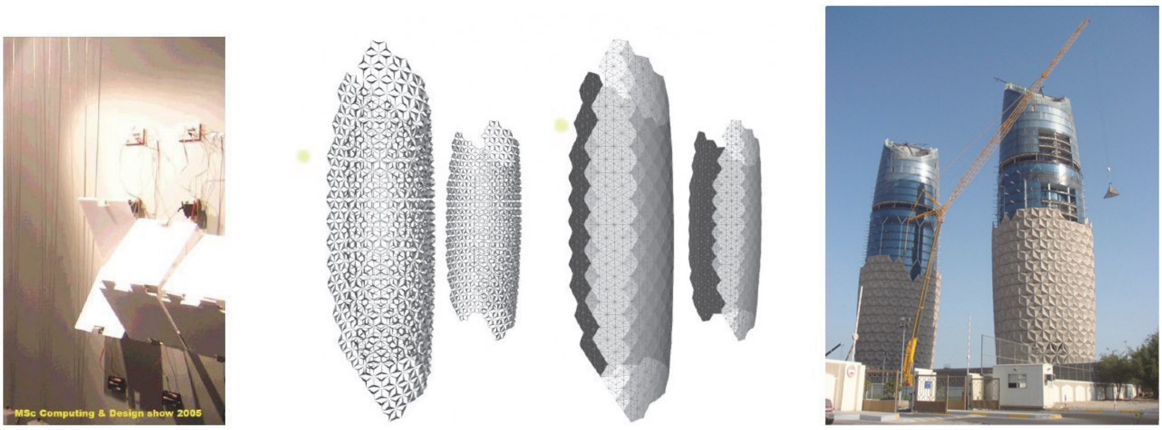


Fig334. *Typical development from academic R&D to realization: (left) 2004 workshop about responsive structures at the CECA with student Abdulmajed Karanouh; (middle) who was employed at Aedas and together with CDR developed the Al Bahr towers competition in 2007 based on the UEL workshop; and (right) construction on site in 2013, Abu Dhabi by Aedas with Arup; an implementation cycle of 9 years is short and Al Bahr is now perceived as a prototype typology for towers in arid regions (Oborn 2013)*

Apart from a lack of theoretical and technical knowledge allocated at universities in this field, there are two other key issues driving industry: *value-addition* and *visibility*. Innovation either produces a value-addition such as procurement, cost or

[102] *"We try to avoid building-in theoretical attitudes, and to reduce the semantic content of our systems to a low level on the basis that flexibility and intelligence are inversely related; and that flexibility is more important."* (Richens 1994, p305)

time efficiencies; or it increases visibility as marketing efficiency. The reputation of Computational Design has been greatly damaged by highly visible academic developments such as the AA Design Research Lab's attempts at 'surface morphogenesis' (Hensel et al 2004) or Greg Lynn's Embryological House around 2005[103]. The application of algorithms for purely formal reasons without spatial or planning logic only advertised marketing in efficiency to the profession and made professionals sceptical, delaying a proper debate about its use: *"[…] it is cheating to muck around with algorithms and mapping programs to generate facade details, as some modish architects do."* (Pearman 2005)

Yet, added planning and design value is finally being noticed and breaking into main-stream profession. Many masterplans and urban design aspects are investigated through computational design, of which Euston Crossing in 7.2 or EXPO 17 in 8.2 are only two examples. Architectural design is starting to apply algorithmic techniques for spatial planning at several larger firms and will set a trend that took 20 years to mature. Most methods in the profession are based on generalized meta-heuristics for routine design. But where is the distinction between *routine* and *creative* design? Is the design of 'good' circulation possibly not the most creative aspect of a building by worrying the *generic function* and thus the experience of use?

To conclude on Applicability, evolutionary optimization is used to illustrate a 20 year timeline of implementation from academic research to commercial software (from personal experience):

- 1995 Frazer and Coates introduce evolutionary algorithms (EA) into architectural design
- 2000 Coates teaches members of CDR
- 2004 introduction to profession: Faulty Towers by author and Frazer's collaboration with S333 (Moller 2005)
- 2006 Pareto optimization for ENK + later SSSP
- 2008 extensive use of EA projects incl. the Khalifa-bin-Zayed competition
- 2008 invitation to demonstrate projects at Autodesk in Las Vegas where CEO Karl Bass announces algorithmic design as new business objective
- 2008- several EA projects for space planning at Aedas/ WoodsBagot
- 2014 introduction of EA as plug-in to AutoCAD (Project Dreamcatcher)[104]
  ➔ Commercially generalized for mainstream application

[103] http://www.glform.com/embryonic/embryonic.htm, accessed 11.01.2015
[104] http://autodeskresearch.com/projects/dreamcatcher, accessed 11.01.2015

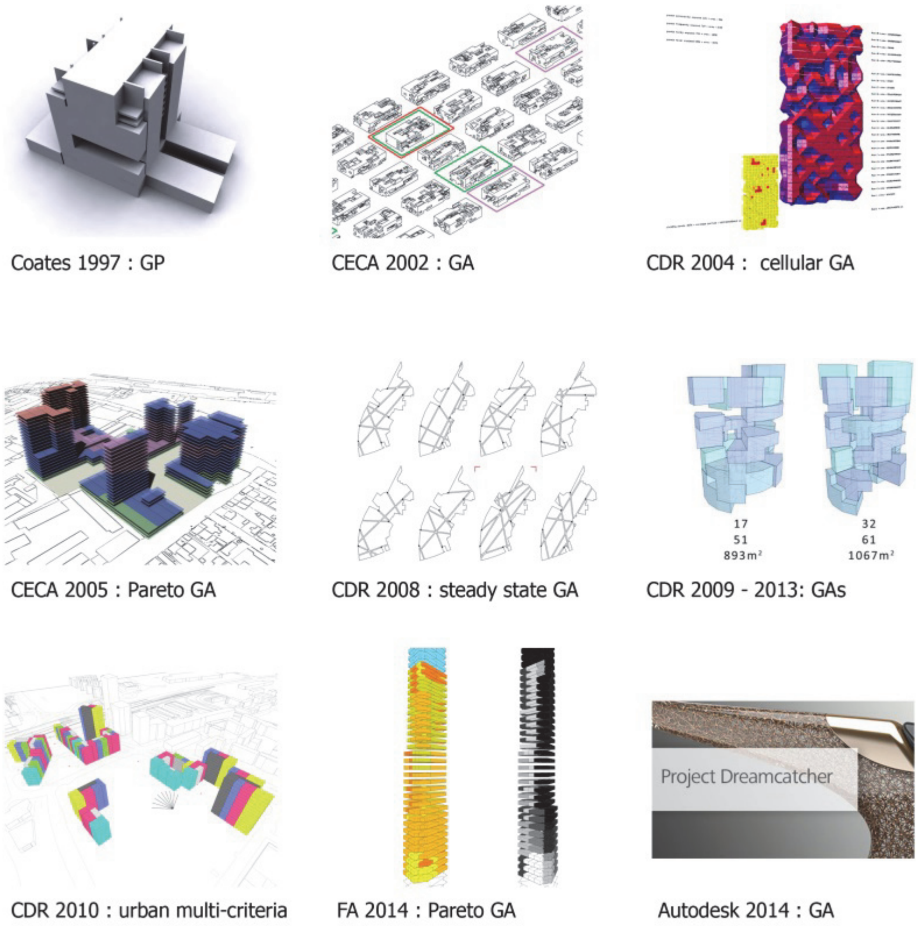


***Fig335**. R&D development of Evolutionary Algorithms (EA) in architectural design (from author's perspective): mid-90s Frazer and Coates explore EA (Frazer more GA, Coates more Genetic Programming (GP)); CECA standard teaching GAs 1995 – 2009; CDR develops many different GA types combined with other algorithms 2004 – now (as WB SUPERSPACE); Autodesk finally present Project Dreamcatcher for the commercial market in 2014*

9.4.3 Deliverables Now and Then

The organization of the architectural workflow into stages has evolved to reflect the methodology architects use and deliverables exchanged with other stakeholders. Computation permeates all architectural design by now, mainly in the form of CAD. But the workflow organization and its deliverables have not been updated to mirror new methodologies. This might have various reasons, including slow legislative proceedings, but a key factor points towards commercial software developed *for* architects. The aim of commercial software developers such as Autodesk or Bentley is to sell licences, not to innovate new methodology. Hence, CAD and peripheral analysis simulation such as climatic or structural evaluation follows existing design workflows, matching computational output formats to traditional deliverables. Software is created to increase efficiency by automating processes and existing knowledge, inheriting the paradigm of the parametric KbD community. BIM is the most advanced development of efficient workflow automation, making clients demand architects to use BIM software for budgetary reasons. It appears no coincidence then that BIM is the only new computational paradigm adopted by the RIBA new Plan of Work.

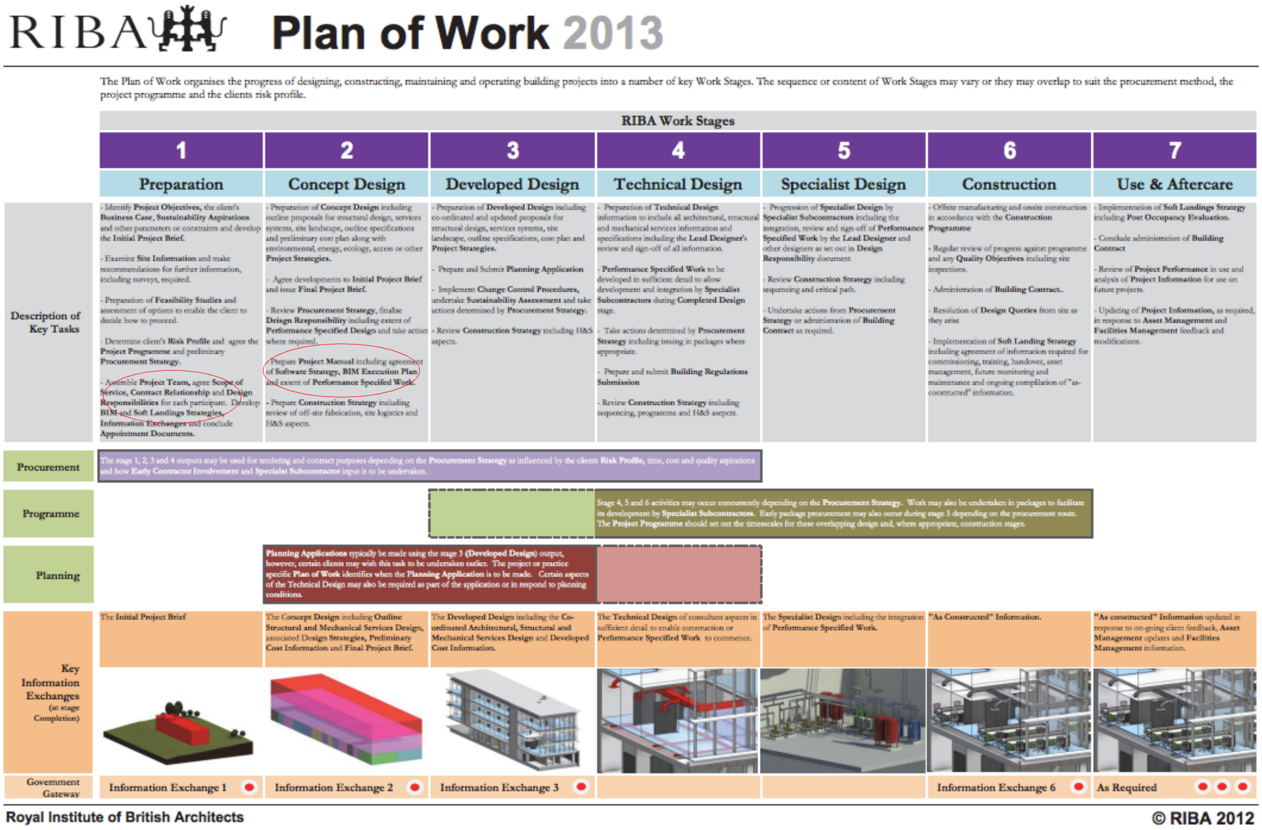


***Fig336**. Royal Institute of British Architects (RIBA), Plan of Work, 2013: only stage 1 & 2identify the development of BIM strategies (red ellipses) but no further digital components*[105]

It becomes evident from collaborative projects such as SSSP's Urban Mix and Density (UMD – see 7.1) that new stages and process phases could be introduced if generalized computational meta-heuristic methodologies were applied. Planners of Tower Hamlets and Newham realized that the old distinction into zoning plans established for masterplans with stages of mix followed by scale, could be reviewed to allow for earlier scrutiny of more complex schedules and distributions rather than broad-brush zoning plans engendering segregated plot developments. The use of methods like UMD shown also in the EXPO 17 and applied in many other Aedas projects, usually results in the omission of zoning plans in favor of continuous distributions across sites and isomorphic density and scale treatments.

Similarly for layout design, the current process includes seven phases: (0) generate envelope by sketch, constraints or both (1) schedule (= approximate capacity), (2) bubble diagrams for coarse room groups (= zoning), (3) scale to schematic geometric layout, (4) develop unit layouts, (5) fill into schematic layout and (6) check capacity and adjust. There are several deviation when using OFSS instances: a) no sketch or envelope approximation until (6); (2,3,5) are generated concurrently through interactive and iterative enquiry, producing (0,1). Instead, (4) is conducted by analogue means in parallel to (2,3,5), leading to a convergence in (6) and reducing the workflow to four instead of seven phases. Complexity is generated earlier in place of formal analogies.

105 Retrieved from http://www.ribaplanofwork.com, accessed 12.01.2015

9.5 CONCEPTUAL REMARKS

Finally, some concluding remarks regarding conceptual aspects of this dissertation that are not discussed in any depth as they would constitute new R&D strands.

9.5.1 Correlations

The dissertation frequently employs the concept of *correlations* between two or more aspects: between analogue heuristics and algorithmic syntax, between algorithmic structures, between occupant affordances and computational analysis etc. Some pertinent correlations that have not been discussed within the case studies are summarized here.

CORRELATIONS BETWEEN DESIGN HEURISTICS & ALGORITHMS

As elaborated in chapter 2, early computational design following the cybernetic paradigm aimed at finding exact algorithms for specific case based problems. All algorithms were heuristics in that sense for automating particular procedures or knowledge schemata. The second-order cybernetics of the New Epistemologists aimed at finding *designerly* phenotypes of spatial environments encoded in the genotypic epistemology of meta-heuristic algorithms. Section 1 of John Frazer's book, "New Tools" (Frazer 1995, pp23-64) provides an early categorization of meta-heuristic algorithms for *designing* or rather *generating* architectural environments. Frazer's projects demonstrated some relations that might have influenced later developments in the field, such as Cellular Automata being used for partitioned topological arrangements, Genetic Algorithms for evolving geometric form and Neural Network analogies to recognize phenomena like symmetry.

Correlations between design heuristics and computational meta-heuristic used in this dissertation are discussed in the conclusions of chapter 5 and introductions of chapter 6.3 and 7.3. Similar to Frazer's (1995) selection of algorithms, no specific design typological heuristic can be mapped into algorithms and vice versa, rather a strategic design methodology can be mapped such as

- Evolutionary Algorithms supporting parametric search and combinatorial optimization,
- Cellular Automata facilitating topological aggregations,
- Agent-based Systems including physical force simulations enabling several searches such as topological configuration of geometric organization ,
- Graph and Network transforms providing configurational control or
- Unsupervised Neural Networks generating tactical associations such as feature classification of design space to define strategies

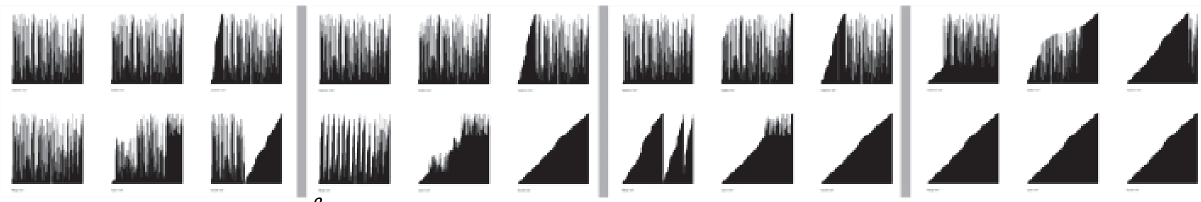


***Fig337**. Sorting algorithms, Åsmund Izaki, 2012: direct correlation between effects of behavioral efficiency and aesthetics of different heuristics of sorting algorithms; figure shows four stages of six sorting heuristics (selection, bubble, insertion, merge, quick and bucket); lecture given at TU Munich, January 2012*

It is easier to categorize individual globally constrained generative algorithms as those listed above to an aspect of design than it is to map associative structures of a USOM into design heuristics. Åsmund Izaki of SUPERSPACE occasionally opens lectures by demonstrating that *sorting* algorithms as elemental algorithmic heuristics (Fig337) could be compared to what in architecture is perceived as a fundamental design principle such as Le Corbusier's five points that lead to some spatial expressions like the Domino house. Izaki compares an algorithmic principle to design principles rather than matching exact computational and design methods.

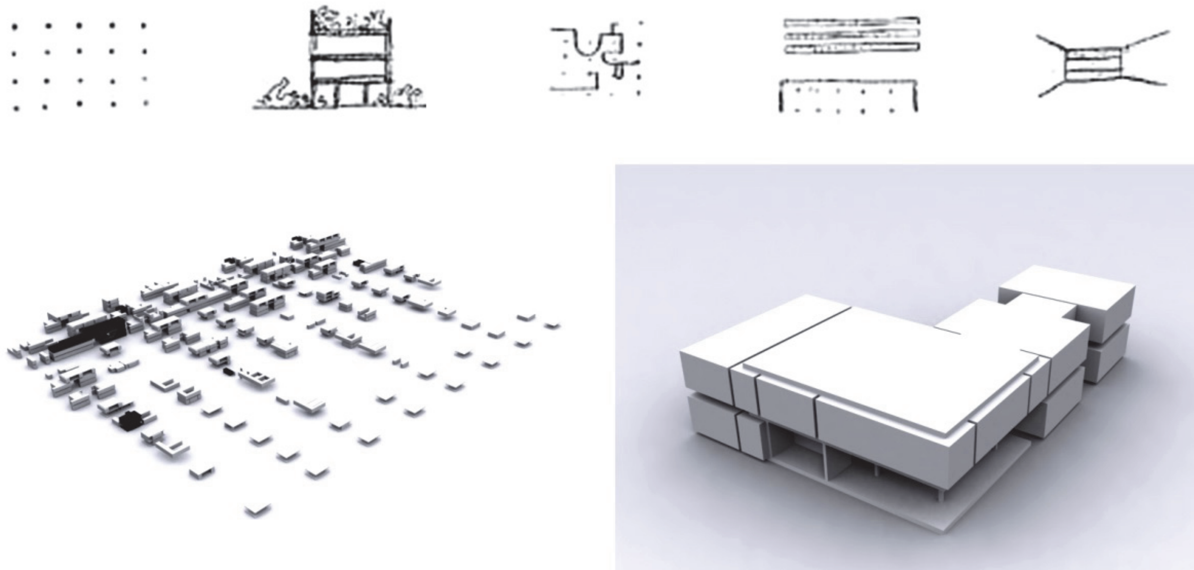


Fig338. *Le Corbusier's 'five points' of architectural composition (above): pilotis, roof garden, free plan, strip windows and curtain wall (Le Corbusier 1985); (below) Le Corbusier's five points do not represent a heuristic but a catalogue that can be assembled in many ways (with however few permissible permutations) as demonstrated by the genetic programming project by Paul Coates (Coates and Hazarika 1999)*

CORRELATIONS BETWEEN SPACES AND ALGORITHMS

Even more elusive than correlations between design heuristics and algorithmic meta-heuristics are spatial configurations in relation to syntactical structure of algorithms. There are no standard categories of phenomenological or spatial knowledge units other than typological distinctions like rooms, corridors (streets) and squares (enclosed/ convex) as proposed by Lynch (1960) or Alexander (Alexander et al. 1977), or sector-based typologies of building logic such as perennially listed in Metric Handbooks like the Metric Handbook (Littlefield 2008). Philip Steadman's (1998) research into archetypes of building form represents an attempt at finding representations of simplified rectangular partitions into building typologies constrained by basic parameters such as light and access.

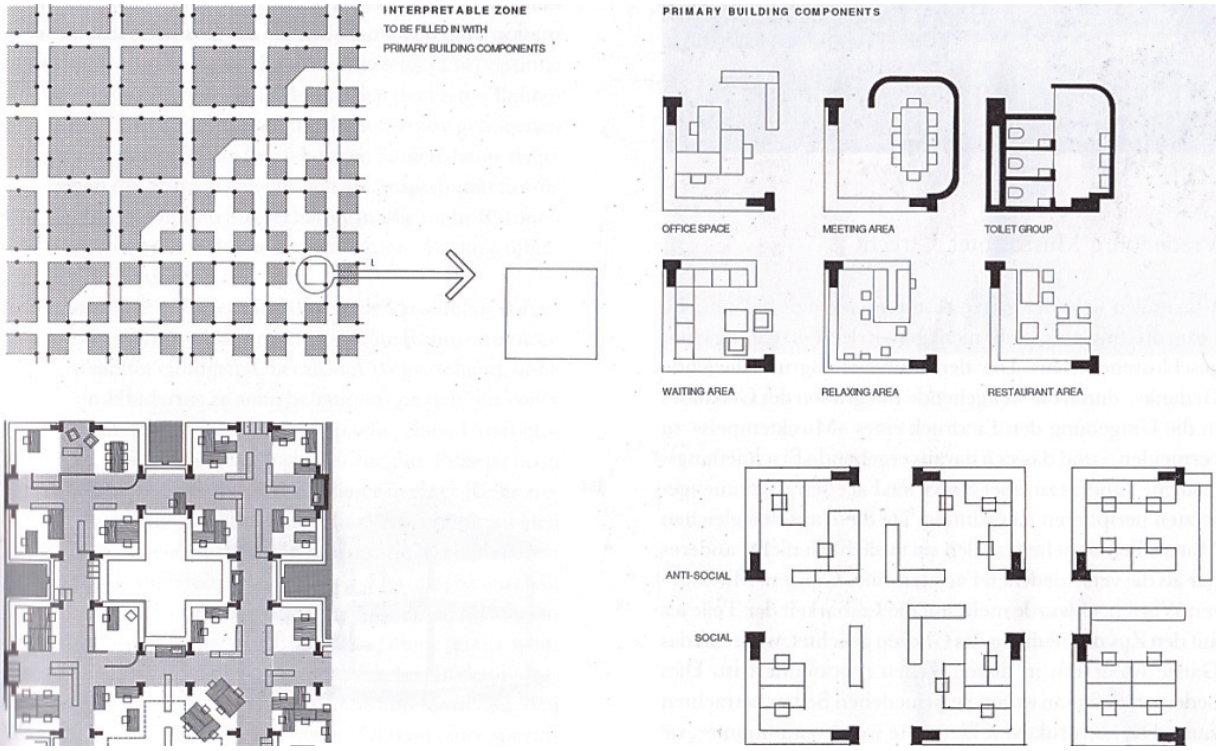


***Fig339**. Herman Hertzberger, Centraal Beheer, 1968: Hertzberger combined rigorous rule-based design with user-behaviour classes, creating correlations between potential user occupation and geometric permutations (Hertzberger 2005)*

Hillier proposed that *generic functions* encoded in spatial configurations are based on levels of movement. All spaces can be reduced to continuous and discontinuous spaces, with continuous permeable complex (p-complex) determining experience of use (Hillier et al. 1976). Like Steadman, Hillier points out that strongly programmed spaces harbour little generative capacity (Hillier 1996, pp197-201). This can be interpreted for both the design generation and for affordances of occupancy.

Scharoun and Hertzberger establish correlations between spatial structures and user occupation affordances but are careful not to encode those explicitly. Instead Scharoun (Janofske 1984) talks of *improvisations* as a design heuristic to approximate associations between space and user. Similarly, Hertzberger (2014) develops catalogues of discrete spatial structures that when assembled through a contextual heuristic generate spaces of *polyvalences*, meaning multiple behavioural and social affordances.

Stan Allen attempted to relate computer-based operations to architectonic patterns of buildings and cities in his essay From Object to Field (Allen 1997). Repetitive series of local transformations give rise to 'field conditions' that abandon classical architectural typologies. Allen excluded the 'user' in his writings and focussed on perceived graphic patterns rather than actions and behaviours and limited his theory of the field to parametric geometric transformations.

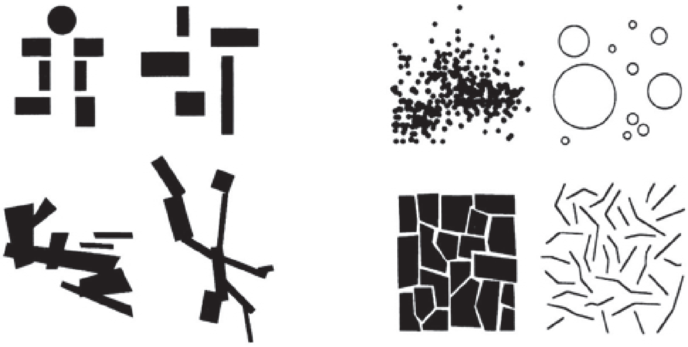


***Fig340**. Organizational Strategies, Stan Allen (1997): from classical organizations (left) to field conditions (right)*

In his essay Programs as Paradigms, Pablo Miranda proposes that the construction of the computer itself limits programmes that an algorithm can provide (Miranda 2014). He proposes that due to its serial nature only certain data structures are possible that constrain any algorithmic epistemology. A list of basic algorithms is proposed that provide the basis of most architectural computing, including graphs and particle-based representations. It is suggested that their expression represents certain programmes.

Most considerations of computational algorithms relating to architecture seem to reflect on the resulting graphic pattern as the visual abstracted constant of architectural space. Yet it is the use of space, the behaviour and cognitive process during design that generates patterns. This dissertation and the OFSS attempt to approximate this correlation that Hillier already anticipated 40 years ago by not focussing on specific algorithms, data structures, spatial environments or design heuristics. While correlations remain vague, the aim has been to develop a system that does not focus on particular programmes but on the meta-structures and behaviours that can give rise to programmes based on experiencing space and use. Correlation tables at the end of sections in chapter six attempt to find loose co-relations between algorithmic representations and behavioural or cognitive affordances, describing the use of spatial environments. Allen pre-empted his limitation of graphic patterns by speculating that only by understanding the behaviours of users will field conditions be understood, potentially giving rise to new methodologies for designing space (Allen 1997, p26; see introductory quote 8.3).

9.5.2 Archetypes of Use vs Typologies of Buildings

"Generic function refers […] to aspects of human occupancy of buildings that are prior to any of these: to occupy space means to be aware of the relationships of space to others, that to occupy a building means to move about in it, and to move about in a building depends on being able to retain an intelligible picture of it." (Hillier 1996, p284)

Many quotes have been employed to demonstrate a certain consensus between architectural researchers (Steadman, Hillier), theorists (Schmarsow, Papert) and practitioners (Allen, Hertzberger) that the use and experience of space should be the main driver for the design of spatial environments. In the conclusion of Space is the Machine, Hillier summarized how a building design process is ideally structured by a three-step hierarchy of *filters of purpose* (Hillier 1996, p300):

1. defining the Generic Function
2. identifying Cultural Intent
3. applying typological Building Differences

OFSS instances such as *FuCon* and design studies such as *Floating Room* translate the *filters of purpose* and start with the movement experience before applying regulatory constraints from typological standards. According to the introductory quote to this section, this would therefore mean that an abstraction of the *intelligible picture* of the occupation of a building into a generative framework has been established.

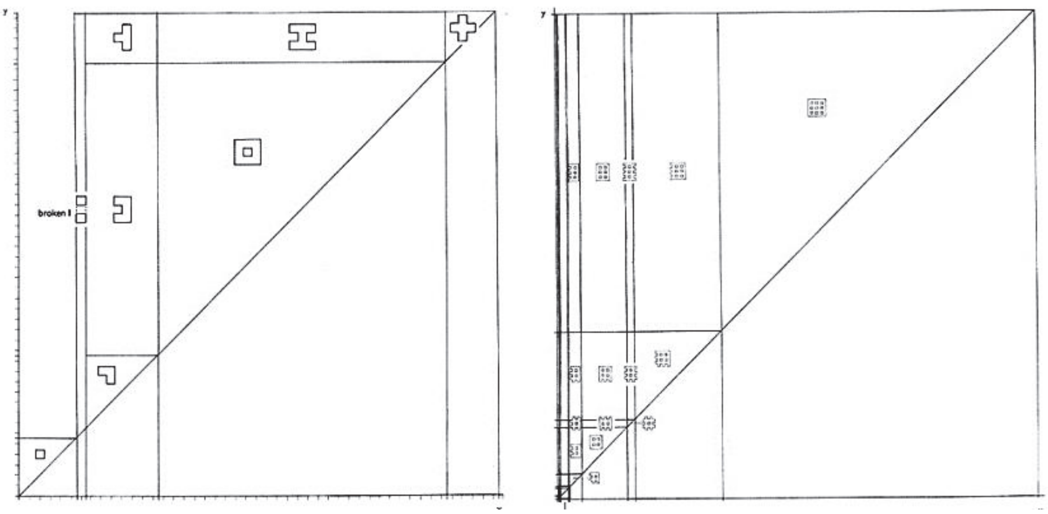


Fig341. *Philip Steadman (2014): (left) simplest plans as basic forms derived from a building 'archetype' are clustered by x-and y-axis binary cells akin to a biological morphospace and serve as 'generators' ;(right) building plans constrained by elementary functions such as light produced from the 'generators' are clustered into classes of built form*

The proposed hierarchy of design purpose filters implicitly suggests that sector-based specifications of typologies should be secondary to experience-based descriptions. That typology might be a definition from experiences based on corporeal use as in Papert's *body-syntonicity* (1980) rather than logistical use as in architectural programmes. Typology adheres to sector-based specifications of building organizations according to logistical use. What could a classification of experience-based occupation affordances for spatial configurations be defined? Steadman's use of the concept of *archetype* (1998) comes to mind exploring classes of permissible configurations of rectangular building forms. Steadman's archetype increases the abstraction of rectangular buildings from room units to floors configurations and encodes use affordances implicitly via partitions that are informed by light conditions. From this archetypal abstraction, a series of generator partitions are extracted that he claims can encode most of architecture's historic permissible (and functional) configurations. He called the classification *morphospace* as an analogy to classifications of dimensional transformations of biological forms (Steadman and Mitchell 2010; Steadman 2014). Extending Steadman's notion of archetype from formal into experiential organizations, a new classification of spatial environments into patterns in-use offers itself[106].

[106] Sean Hanna (2007b) at UCL conducted a classification of configurations to approximate stylistic archetypes, applying however the traditional separation of analysis and genesis.

***Fig342**. Three building types whose design paradigm has been converted from efficient space packing to human-centric affordances: (left) underground station such as Westminster, London (Hopkins architects, 1999) need to plan for visible flows as if swarming towards daylight; (middle) schools such as the Peterborough Deacon Academy (Foster architects, 2007) plan for passive supervision and mutual motivation; (right) workplaces such as the Macquarie HQ London (Clive Wilkinson architects, 2011) increase circulation to encourage indirect communication (physical encounters) for higher productivity (BCO 2009; Sailer et al. 2010)*

Now that the 'tools' are beginning to be available to understand patterns of user behaviour and perception in correlation to spatial form, Allen's speculation of a new design methodology to model field conditions from field experiences appears tangible. The briefing for a building project should start with the specification of an archetype of experience rather than an architectural form to be generated, scoping correlations between generic functions and spatial configurations. Current trends in building sectors support this approach as increasing numbers of building typologies are focussing on defining spatial or social experiences before allocating functional areas (Derix 2014; Derix and Jagannath 2014). If one was to follow Steadman's approach, which proposes to generate classifications of configurations from which the designer-observer can choose (Steadman 2014), then an associative search model based on self-organizing neural networks as presented before seems plausible since they allow for disparate qualitative parameters to be classified statistically that open the design search beyond geometric performances. Further, self-organizing ANNs generate the complete *morphospace* from given input samples from which the designer as observer can explore non-formal dissections with their associative weighting, that is the designer will learn which spatial schemata can occur and how they are composed. This *morphospace* would have to be constrained by simultaneous analytical evaluation to reduce the field to permissible phenotypes as done by Steadman and demonstrated by models of the OFSS.

In *Practice*, Stan Allen (2008) instrumentalizes the philosopher Nelson Goodman's distinction into auto- and allographic arts, where autographic arts represent expressions of personal conventions like architectural diagrams and allographic arts represent ephemeral expressions from abstracted time-based conventions like musical scores. Allen suggests that computer algorithms are akin to allographic notation, allowing others to regenerate empathic instances of encoded phenomena. Also Seymour Papert's proposition of algorithms as heuristic process to learn "*to achieve a direct aesthetic experience through mathematics*" (Papert 1980, p118) recalls August Schmarsow's (1894) observation that aesthetics is an act of empathy, acquired through experiencing space (Derix 2014b). This dissertation and the development of the OFSS with its on-going instantiations represent an attempt to translate the original theory of organic architecture via computational algorithms into a design context.

10 BIBLIOGRAPHY

ACPO (Association of Chief Police Officers). *Secured by Design – New Schools 2014.* www.securedbydesign.com: accessed 02.09.2014

Aedas|R&D (2007). *Aedas|R&D World Trade Centre Memorial Museum - Computational Analysis & Development Report*. Project report, submitted to DavidBrodyBond, July 2007

Aedas|R&D (2013). *Packages Retail Centre Spatial Analysis report*. Project report, submitted to Aedas Birmingham, May 2013

Aichholzer O and Aurenhammer F (1996). "Straight skeletons for general polygonal figures in the plane"*.* In Proceedings of the 2nd Internatinoal Conference for Computing and Combinatorics (COCOON '96), *Lecture Notes in Computer Science*, no. 1090, Springer: 117–126

Akin O, Dave B, and Pithavadian S (1992). "Heuristic generation of layouts (HeGeL): based on a paradigm for problem structuring". *Environment and Planning B,* vol. 19: 33 - 59

Akin O (1998). *Cognition-based Computational Design*. In Proceedings of the 30th Anniversary Celebration meeting of The Key Center for Design Computing, University of Sydney, Sydney

Alexander C (1964). *Notes on the Synthesis of Form.* Harvard University Press, Cambridge MA

Alexander C (1968). "Systems Generating Systems"*. Architectural Design*, vol. 38, Wiley & Sons, London: 605-610

Alexander C, Ishikawa S and Silverstein M (1977). *A Pattern Language – Towns Buildings Construction.* Oxford University Press, London

Allen S (1997). "From Object to Field". *Architectural Design - Architecture after Geometry*, no. 127, John Wiley & Sons, London: 24-31

Allen S (2008). *Practice: Architecture, Technique and Presentation.* Routledge, London/New York

Arnheim R (1974). *Art and Visual Perception – a psychology of the creative eye.* University of California Press, London; (extended version – original 1954)

Arthur P and Passini R (2002). *Wayfinding: People, Signs, and Architecture.* McGraw-Hill, Ontario, 1992; (original reissued as a collector's edition by Focus Strategic Communications)

Arvin S and House D (2002). "Modeling architectural design objectives in physically based space planning"*. Automation in Construction*, vol. 11, Elsevier: 213-225

Barnes A and Thagard PR (1997). "Empathy and Analogy". *Dialogue*, vol. 36, issue 4, Cambridge University Press: 705-720

Barthes R (1977): *Image, Music, Text*. Heath, S (trans.), Fontana Press, London

Batty M and Longley P (1994). *Fractal Cities – A Geometry of Form and Function.* Academic Press, London

Batty M (2004). *A New Theory of Space Syntax.* Working Papers Series, ISSN: 1467-1298: http://www.casa.ucl.ac.uk/working_papers/paper75.pdf, accessed march 2010

Batty M and Rana S (2004): "The automatic definition and generation of axial lines and axial maps"*. Environment and Planning B*, vol. 31, issue 4: 615–640

Beauchemin S and Barron J (1995). "The Computation of Optical Flow". In *ACM Computing Surveys*, vol. 27, issue 3: 433-466

Benedikt ML: "To take hold of space: isovists and isovist fields"*. Environment and Planning B*, vol. 6, 1979: 47 – 65

Benoudjit A, Derix C and Coates P (2004). *Human perception and space classification: The Perceptive Network*. In proceedings of Generative Arts conference, Milan

Berghauser Pont M and Haupt P (2004): *Spacemate: the Spatial Logic of Urban Density*. DUP Science, Delft

Blum H (1967): "A Transformation for Extracting New Descriptors of Shape". *Pattern Recognition*, vol 10:176-180

Blum C and Blesa M (2005): "New meta-heuristic approaches for the edge-weighted K-cardinality tree problem". *Computers & Operations Research*, vol 32, issue 6, Elsevier, Oxford: 1355 – 1377

Bogner D (2001): *Frederick Kiesler - Endless Space.* Hatje Cantz publishers, Ostfildern-Ruit

Bonabeau E, Dorigo M and Theraulaz G (1999): *Swarm Intelligence*. Oxford University Press, New York

Bourke P (1994): *Polygonising a scalar field*. http://paulbourke.net/geometry/polygonise/, accessed 2000

Braach M (2014). "Solutions You Can't Draw"*.* In Derix C and Izaki A (eds.), *Empathic Space: The Computation of Human-Centric Architecture*. Architectural Design, Wiley & Sons, London/New York: 46 – 53

Branzi A (2006). *No-Stop City – Archizoom Associati*. HYX, Orleans

Braitenberg V (1984). *Vehicles – Experiments in Synthetic Psychology.* MIT Press, Cambridge MA

British Council for Offices (BCO) (2009): *Guide 2009 – Best Practice in the Specification for Offices*. London

Bruls M, Huizing K and van Wijk, JJ (2000). *Squarified Treemaps*. In proceedings of Joint Eurographics and IEEE TCVG Symposium on Data Visualization, Springer, Heidelberg New York: 33–42

Bullock N, Dickens P and Steadman P (1972)." The Modelling of Day to Day Activities". In Martin L and March L (eds.), *Urban Space and Structure*, Cambridge University Press: 129-158

Cariani P (1993)." To Evolve an Ear: Epistemological Implications of Gordon Pask's Electrochemical Devices". *Systems Research*, vol 10 (3): 19-33

Chiaradia A, Hillier B, Schwander C and Wedderburn M (2009): Spatial Centrality, Economic Vitality/Viability - Compositional and Spatial Effects in Greater London. In Proceedings of 7th Space Syntax Symposium, Stockholm: 016.1-19

Cilliers P (1998): *Complexity and Post-Moderism – Understanding Complex Systems.* Routledge, New York

Coates P, Healy N, Lamb C and Voon WL (1996). *The Use of Cellular Automata to Explore Bottom-Up Architectonic rules*. In proceedings of Eurographics UK chapter 14th Annual Conference: 26-28

Coates P and Hazarika L (1999). *The Use Of Genetic Programming For Applications In The Field Of Spatial Composition*. In Proceedings of the 2nd Generative Arts conference, Milan

Coates P and Schmid C (1999): *Agent-based Modelling.* In proceedings of eCAADe, *Architectural Computing: from Turing to 2000*, CAADRU, Liverpool

Coates P, Derix C, Simon C and Appels T (2001). *Current work at CECA. Three projects: Dust, Plates & Blobs*. In proceedings of the 3rd Generative Arts conference, Milan

Coates P (2004). *Review Paper: Some Experiments Using Agent Modelling At CECA*. In proceedings of the 7th Generative Arts conference, Milan

Coates P, Derix C, Parvin T, Puusepp R and Lau T (2005). *Topological Approximations for Spatial Representations.* In proceedings of the 8th Generative Arts conference, Milan

Coates P, Thum R and Derix C (2006). *The VIPA project - some notes on the pedagogical approach to design education using active 3d worlds.* In proceedings of the 24th eCAADe conference, Liubljiana

Coates P and Derix C (2007). *Parsimonious Models of Urban space*. In Proceedings of the 25th European Conference for Computer Aided Design in Architecture Education (eCAADe) conference, Frankfurt: 335-342

Coates P and Derix C (2008). *Smart Solutions for Spatial Planning - A Design Support System for Urban Generative Design*. In Proceedings of the eCAADe, Antwerp

Coates P (2010). *Programming.Architecture*. Routledge, London

Coates P and Derix C (2014). "Deep Structure of the Picturesque". In Derix C and Izaki Å (eds.), *Empathic Space – the Computational of Human-centric Architecture,* Architectural Design, vol. 85, issue 2, Wiley and Sons, London: 32-37

Commission for Architecture & the Built Environment (CABE) (2000). *By Design: Urban design in the planning system: towards better practice.* Department for the Environment, Transport and the Regions, London

Chomsky N (1989). *Syntactic Structures.* Mouton, Den Haag, 1957 (first edition); de Gruyter, Berlin/New York (second edition)

Colorni A, Dorigo M and Maniezzo V (1991). *Distributed Optimization by Ant Colonies*. In Proceedings of the First European Conference on Artificial Life, Elsevier Publishing: 134-142

Conroy-Dalton RA (2003). "The secret is to follow your nose: Route path selection and angularity". *Environment and Behaviour*, vol. 35: 107–131.

Corne DW, Knowles JD and Oates MJ (2000). *The Pareto Envelope based Selection Algorithm for Multi-objective Optimization*. In proceedings of the Fourth Parallel Problem Solving from Nature conference: 839-848.

Corner J (1999). "The Agency of Mapping – Speculation, Critique and Invention*".* In Cosgrove D (ed.), *Mappings.* Reaktion Books, London: 231-252

Coyne R (1988). *Logic Models of Design*. Pitman, London

Coyne R, Rosenman MA, Radford AD, Balachandran M and Gero JS (1989). *Knowledge-Based Design Systems*. Addison-Wesley Longman Publishing, Boston, MA

Coyne R (1990). "Design Reasoning Without Explanations". *Artificial Intelligence Magazine*, vol. 11, no. 4: 72-80

Coyne R and Newton S (1990). "Design Reasoning by Association". *Environment and Planning B*, vol. 17: 39-66

Coyne R and Postmus A (1990). "Spatial Applications of Neural Networks in Computer-Aided Design". In *Artificial Intelligence in Engineering,* vol. 5, no. 1: 9-22

Coyne R (2011). *Derrida for Architects.* Routledge, London/New York

Crooks A, Castle C and Batty M (2008). "Key Challenges in Agent-based Modelling for Geo-Spatial Simulation". *Computers, Environment and Urban Systems* 32: 417–430

Crucitti P, Latora V and Porta S (2006). "Centrality in Networks for Urban Streets*".* In *Chaos – Interdisciplinary Journal of Non-linear Science*, vol. 16, issue 1, American Institute of Physics

Cruise RB (2005). "The Dry Stone Waller". In Lim CJ (ed.), *Devices*, Architectural Press, London: 188 – 193

Darke J (1979). "The Primary Generator and the Design Process"*. Design Studies*, vol. 1, no 1: 36-44

Das A and Chakrabarti B (eds.) (2005). "Quantum Annealing and Related Optimization Methods". *Lecture Notes in Physics*, vol. 679, Springer, Heidelberg

De Berg M, van Krefeld M, Overmars M and Scharzkopf O (1997): *Computational Geometry – Algorithms and Applications.* Springer, Berlin-Heidelberg

De Kestelier X and Peters B (eds.) (2013). "Computational Works – the Building of Algorithmic Thought". *Architectural Design*, profile 222, March/April

Delaunay B (1934). "Sur la sphère vide". *Izvestia Akademii Nauk SSSR*. Otdelenie Matematicheskikh i Estestvennykh, Nauk 7: 793–800

Deleuran AH and Derix C (2013). *Topological Infrastructure Analysis of the Built Environment.* In Proceedings of the 11th European Architecture Envisioning Association (EAEA) conference, Milan: 419 - 426

Derix C and Thum P (2000). *Self-Organizing Space*. In proceedings of 2nd Generative Arts conference, Milan

Derix C (2001). *SOS Self-Organizing Space*. MSc thesis, University of East London, submitted October 2001

Derix C (2004). *Building a Synthetic Cognizer*. In proceedings of the Design Computation Cognition conference, Boston MA

Derix C (2006). "Approximating Phenomenological Space*"*. In Proceedings of Intelligent Computing in Engineering and Architecture, *Lecture Notes in Artificial Intelligence*, Springer, Heidelberg: 136–46

Derix C (2008). "Genetically Modified Spaces". In David Littlefield (ed.), *Space Craft: Advanced in Architectural Computing*, RIBA publications: 48-57

Derix C, Miranda P and Gamlesaeter Å (2008). *3d isovists and spatial sensations: two methods and a case study.* In Proceedings of EDRAMOVE & SFB TR8 conference, VeraCruz, Mexico: 67-72

Derix C (2010). *Mediating Spatial Phenomena through Computational Heuristics.* In proceedings of the 30th American Computer-Aided Design in Architecture (ACADIA) conference, New York: 61-66

Derix C, Gamlesaeter A, Miranda P, Helme L and Kropf K (2012)."Simulation Heuristics For Urban Design". In Mueller-Arisona S , Wonka P, Aschwanden G and Halatsch J (eds.), *Digital Urban Modelling and Simulation,* Communications in Computer and Information Science (CCIS), vol. 242, Springer, Berlin Heidelberg: 165-186

Derix C (2012a). "Digital Masterplanning: Computing Urban Design"*.* In *Urban Design and Planning*, Institution of Civil Engineers, vol. 165, issue 1, Thomas Telford: 1-15

Derix C (2012b): *Implicit Space.* In Proceedings of the 39th eCAADe conference, Prague: 583-591

Derix C and Gamlesaeter A (2012). "Behavioural Prototypes in Architectural Design Computation"*.* In Petruschka J and Adenaur J (eds.), *Prototype! Physical, Virtual, Hybrid, Smart: Tackling New Challenges in Design & Engineering*. Form + Zweck, Berlin: 124-140

Derix C and Izaki Å (2013). "Spatial Computing For the New Organic". In De Kestelier X and Peters B (eds.), *Computational Works – The Building of Algorithmic Thought*. Architectural Design, profile 222, Wiley and Sons: 40-45

Derix C (2014a). "The Space of People in Computation". In Derix C and Izaki Å (eds.), *Empathic Space – the Computational of Human-centric Architecture.* Architectural Design, vol. 85, issue 2, Wiley and Sons, London: 14-23

Derix C (2014b). "Space Planning with Synthetic User Experience". In Andia A and Spiegelhalter T (eds.), *Post-Parametric Automation in Design and Construction*, Artech House, Norwood MA

Derix C and Izaki A (2014): "New Curricula – Syntonic systems"*.* In Derix C and Izaki Å (eds.), *Empathic Space – the Computational of Human-centric Architecture.* Architectural Design, vol. 85, issue 2, Wiley and Sons, London: 122-129

Derix C and Jagannath P (2014a): "Near Futures – Associative Archetypes". In Derix C and Izaki Å (eds.), *Empathic Space – the Computational of Human-centric Architecture.* Architectural Design, vol. 85, issue 2, Wiley and Sons, London: 130-135

Derix C and Jagannath P (2014b): "Digital Intuition – Autonomous Classifiers for Spatial Analysis and Generative Design"*. Journal of Space Syntax*, vol. 5, issue 2: 190-215

Derix C and Conroy-Dalton RA (2015 – upcoming): User-Experience Typologies of Space. In Hensel M and Nilsson F (eds.), *The Changing Shape of Practice – Integrating Research and Design in Architectural Practice*, Routledge, London

Derrida J (1982). *Margins of Philosophy*. Harvester Press, Brighton

Dijkstra EW (1959). "A note on two problems in connexion with graphs"*. Numerische Mathematik* 1: 269–271

Doxiadis CA (1972). *Architectural Space in Ancient Greece.* MIT Press, Cambridge MA

Eades P (1984). "A heuristic for graph drawing". *Congressus Numerantium* 42 (11): 149–160

Eastman C (1969). *Cognitive processes and ill-defined problems: a case study from design.* In proceedings of the First International Joint-Conference on Artificial Intelligence, Morgan Kaufmann, San Francisco: 669-690

Eisenman P (1963). *The Formal Basis of Modern Architecture*. Ph.D. dissertation, Cambridge University, submitted august 1963; reproduced by Lars Mueller publishers, Baden, Switzerland, 2006

Elezkurtaj E and Franck G (2002). "Algorithmic Support of Creative Architectural Design". In *Umbau* 19, Vienna, 2002: 129-37

Epstein B (2011). Agent-Based Modeling and the Fallacies of Individualism. In Humphreys P and Imbert C (eds.), *Models, Simulations and Representations*, Routledge, New York: 115-144

Evans R (1997). *Translation from Drawing to Building and Other Essays*. Architectural Association Press, London

Fabbri R, Estrozi LF and Da Costa L (2002). "On Voronoi Diagrams and Medial Axes". *Journal of Mathematical Imaging and Vision*, vol. 17, 2002: 27 – 40

Finucane E, Derix C and Coates P (2006). *Evolving urban structures using Computer Optimisation Techniques*. In Proceedings of Generative Art Conference, Milan

Flemming U, Baykan C, Coyne R and Fox M (1992). "Hierarchical Generate and Test vs. Constraint-Directed Search - A Comparison in the Context of Layout Synthesis". In *Artificial Intelligence in Design*, Kluwer Academic Publishers, Dordrecht

Flemming U (1994). "Case-Based Design in the SEED System". Automation in Construction 3, no. 2-3: 123-133

Fortune S (1987). "A Sweep Line Algorithm for Voronoi Diagrams". *Algorithmica,* vol. 2: 153-174

Frampton L (1985). *Modern Architecture: a Critical History.* Thames & Hudson, London

Franz G, Mallot HA and Wiener J (2005). "Graph-based models of space in architecture and cognitive science - a comparative analysis"*. Architecture, Engineering and Construction of Build Environments*, Windsor ON, Canada: 30-38

Franz G and Wiener J (2008). "From space syntax to space semantics: a behaviourally and perceptually oriented methodology for the efficient description of the geometry and topology of environments". *Environment and Planning B*, vol. 35: 574 – 592

Frazer J (1995). *An Evolutionary Architecture*. Architectural Association, London, 1995

Freeman L (1977). "A set of measures of centrality based upon betweenness". Sociometry, vol. 40: 35–41

Fritzke B: "A Growing Neural Gas Network Learns Topologies". In Tesauro et al. (eds.), *Advances in Neural Information Processing Systems*, vol. 7, MIT Press, Cambridge MA

Froehlich J (1996). *Neural Networks with Java*. MSc thesis, University of Applied Science, Regensburg (http://www.nnwj.de – accessed 19.08.1997)

Fussel D and Subramanian KR (1988). *Fast Ray-tracing Using k-d Trees.* Technical report CS-TR-88-07, University of Texas, Austin TX

Gamma E, Helm R, Johnson R and Vlissides J (1995). *Design Patterns – Elements of Reusable Object-Oriented Software.* Addison-Wesley, Boston

Gero J (1990). "Design Prototypes: A Knowledge Representation Schema for Design". *AI magazine*, vol. 11, no 4

Gero JS and Maher ML (eds.) (1993). *Modelling Creativity and Knowledge-based Creative Design*. Lawrence Erlbaum associates, Hillsdale NJ

Gero J and Maher ML (1993). Introduction. In *Modelling Creativity and Knowledge-Based Creative Design.* Gero J and Maher ML (eds.), Lawrence Erlbaum Publishers, Hillsdale NJ: 1-6

Gibson JJ (1950). *The Perception of the Visual World*. Houghton Mifflin

Gibson JJ (1979). *The Ecological Approach to Visual Perception.* Lawrence Erlbaum, 1979

Goldberg DE (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, Reading, Massachusetts

Grossberg S (1976): "Adaptive pattern classification and universal recoding. I. Parallel development and coding of neural feature detectors". *Biological Cybernetics* 23, Springer, Heidelberg: 121-134

Guenzel S (2005). Philisophie und Raeumlichkeit. In *Handbuch Sozialraum*, Kessel, Reutlinger, Maurer and Frey (eds.), Springer VS Verlag, Wiesbaden

Haggett P and Chorley RJ (1969). *Network Analysis in Geography*. Edward Arnold, London

Haken H and Graham R (1971). "Synergetics - The theory of Cooperation". *Umschau* vol. 6

Haken H (2004). *Synergetic Computers and Cognition.* Springer, Berlin

Hanna S (2007a). *Representation And Generation Of Plans Using Graph Spectra*. In Proceedings of the 6th Space Syntax Symposium, Istanbul

Hanna S (2007b). *Automated Representation of Style by Feature Space Archetypes: Distinguishing Spatial Styles from Generative Rules.* International Journal of Architectural Computing, vol.5, issue 1: 2-24

Harding J and Derix C (2010). *Associative Spatial Networks in Architectural Design: Artificial Cognition of Space using Neural Networks with Spectral Graph Theory.* In proceedings of Design Computation & Cognition conference, Stuttgart: 305-323

Hebb DO (1949). *The Organization of Behavior.* Wiley & Sons, New York

Helme L, Derix C and Izaki Å (2014). "Spatial Configuration: Semi-Automatic Methods for Layout Generation in Practice". In Koch and Miranda (eds.), *Models and Diagrams in Architectural Design*, Journal of Space Syntax, vol. 5

Hensel M, Menges A, Weinstock M (eds) (2004). "Emergence – Morphogenetic Design Strategies". *Architectural Design*, vol. 74, no 3, Wiley Academy, London

Hertzberger H (2005). *Lessons for Students in Architecture*. 010 Publisher, Rotterdam

Hertzberger H (2014). "Polyvalence". In Derix C and Izaki Å (eds.), *Empathic Space – the Computational of Human-centric Architecture.* Architectural Design, vol. 85, issue 2, Sept/Oct 2014, Wiley and Sons, London: pp. 106-113

Highsmith J (2000). *Adaptive Software Development: A Collaborative Approach to Managing Complex Systems*. Dorset House Publications, New York

Hillier B and Leaman A (2974). "How is Design possible?" *Journal for Architectural Research and Teaching*, vol. 3, issue 1, RIBA: 4-11

Hillier B, Leaman A, Stansall P and Bedford M (1976). Space Syntax. *Environment and Planning B*, vol. 3: 147-185

Hillier B and Hanson J (1984). *The Social Logic of Space*. Cambridge University Press, Cambridge

Hillier B (1996). *Space is the Machine.* Cambridge University Press

Hillier B, Turner A, Yang T and Park HT (2010). "Metric And Topo-Geometric Properties Of Urban Street Networks: Some convergences, divergences and new results". *Journal of Space Syntax*, vol. 1, no 2: 258 – 279

Hillier B (2014). "The Generic City and its Origins". In Derix C and Izaki A (eds.), *Empathic Space: the Computation of Human-Centric Architecture.* Architectural Design, Wiley and Sons, London: 100-105

Holl S, Pallasmaa J and Perez-Gomez A (1994). *Questions of Perception – Phenomenology of Architecture.* A+U Publishing, Special Issue

Holten D (2006). "Hierarchical Edge Bundles - Visualization of Adjacency Relations in Hierarchical Data". In the Institute of Electrical and Electronics Engineers *Transactions on Visualization and Computer Graphics*, vol. 12, issue 5

Holten D and van Wijk J (2009). *Force-directed Edge Bundlin for Graph Visualization*. Eurographics/ IEEE-VGTC Symposium on Visualization, vol. 28, no 3

Horn J, Nafpliotis N, and Goldberg DE (1994). *A Niched Pareto Genetic Algorithm for Multiobjective Optimization*. In Proceedings of the First IEEE Conference on Computational Intelligence, IEEE World Congress on Computational Intelligence: 82-87

Ireland T and Derix C (2003). *An analysis of the Poly-dimensionality of living - An experiment in the application of 3-dimensional self-organising maps to evolve form.* In proceedings of the 21st eCAADe, Graz: 449-456

Izaki Å and Derix C (2013). *Visibility Polygon Traversal Algorithm*. In Proceedings of the 11th European Architecture Envisioning Association (EAEA) conference, Milan: 427 – 434

Izaki Å and Derix C (2014). "Anticipating Behaviours". In Beesley P (ed.), *Near-Living Architecture - Work in Progress from the Hylozoic Ground Collaboration 2011-2013*. Riverside Architectural Press, Toronto: 73-84

Jabareen YR (2006): "Sustainable Urban Forms: Their Typologies, Models and Concepts". In Journal of Planning Education and Research, vol. 26: 38-52

Janofske E (1984). *Architektur-Raeume: Idee und Gestalt bei Hans Scharoun.* Vieweg & Sohn, Wiesbaden

Jencks C (1971). "Towards the Year 2000". In *Architectural Association Quarterly*, Pergamon Press, Oxford, vol 3, no 1: 56 – 60

Joedicke J (1982). *Das Andere Bauen: Gedanken und Zeichnungen von Hugo Haering*. Karl Kraemer Verlag, Stuttgart

Kalay Y (2004). *Architecture's New Media – Principles, Theories and Methods of Computer-Aided Design*. MIT Press, Cambridge, MA

Kamada T and Kawai S (1989). "An algorithm for drawing general undirected graphs". *Information Processing Letters*, Elsevier, vol. 31 (1): 7–15

Katz P (1993). *New Urbanism – Towards an Architecture of Community.* McGraw-Hill, New York

Keller S (2006). "Fenland Tech – Architectural Science in Postwar Cambridge". *Grey Room* 23, MIT Press Journals, Cambridge MA: 40-65

Keller S (2007). "Systems Aesthetics, or How Cambridge Solved Architecture". In Anstey T, Grillner K and Hughes R (eds.), *Architecture and Authorship*, Black Dog Publishing, London

Kelly GA (1955). *The Psychology of Personal Constructs*. Norton, New York

Kilian A (2006). *Design Exploration through Bidirectional Modeling of Constraints*. PhD thesis, MIT, Boston

Kirkpatrick A, Gelatt CD and Vecchi MP (1983). "Optimization by Simulated Annealing". *Science* 220 (4598): 671–680

Kluetsch C (2007). *Computer Grafik – Aesthetische Experimente zwischen zwei Kulturen. Die Anfaenge der Computerkunst in den 1960er Jahren.* Springer, Wien

Koch D (2012). *Isovists Revisited - Egocentric space, allocentric space, and the logic of the Mannequin*. In proceedings of the Eighth International Space Syntax Symposium, Santiago de Chile

Kodicek D (2005). *Mathematics and Physics for Programmers*. Charles River Media, Hingham MA

Kohonen T (1982): "Self-Organized Formation of Topologically Correct Feature Maps". *Biological Cybernetics* 43, Springer, Heidelberg: 59-69

Kohonen T (1995). *Self-Organizing Maps*. Springer, Berlin Heidelberg

Kolarevic B (2005). "Towards the performative in architecture". In Kolarevic B and Malkawi A (eds.), *Performative Architecture* , Routledge Chapman & Hall, New York

Kraemer J and Kunze JO (2005). *Design Code*. Diploma thesis, Technical University Berlin

Krause D, Derix C, Gamlesaeter A, Poehland K, Braun S and Helme L (2011). *The Virtual Building Simulator – A Post-Parametric Spatial Planning Environment*. In proceedings of the 11th International Conference on Construction Applications of Virtual Reality, Weimar: 447-459

Lang P and Menking W (2003). *Superstudio – Life without Objects.* Skira edizioni, Milano

Langley P, Derix C and Coates P (2007). *Meta-cognitive Mappings – Growing Neural Networks for Generative Urbanism.* In proceedings of the 10th Generative Arts conference, Milan

Langton C (1995). *Artificial Life.* MIT Press, Cambridge MA

Laskari S, Hanna S and Derix C (2008). *Urban identity through quantifiable spatial attributes: Coherence and dispersion of local identity through the automated comparative analysis of building block plans*. In proceedings of the Design Computing and Cognition conference, Springer, Berlin: 615-634

Latour B (1987). *Science in Action - How to follow scientists and engineers through society*. Open University Press, Milton Keynes

Latour B (1999). "The trouble with Actor Network Theory". *Soziale Welt*, vol. 47: 369-381

Le Corbusier J (1985). *Toward a New Architecture*. Dover Publications, New York

Lefebvre H (1991). *The Production of Space*. Donald Nicholson-Smith (trans.), Basil Blackwell, Oxford (originally published 1974)

Leymarie FF and Kimia BB (2008). "From the Infinitely Large to the Infinitely Small – Applications of Meidal Symmetry Representation of Shape". In *Medial Representations*, Computational Imaging and Vision, vol. 37: 327-351

Leymarie FF, Derix C, Miranda P, Coates P and Calderon C (2008). *Medial Representations for driving the Architectural Creative Process*. World Architecture Forum, Barcelona

Liggett R (2000). "Automated Facilities Layout: past, present and future". Automation in *Construction*, vol. 9: 197-215

Littlefield D (ed.) (2008).*Metric Handbook – Planning and Design Data.* Third Edition, Architectural Press, Elsevier, Oxford

Lorensen W and Cline H (1987). "Marching Cubes - A High Resolution 3d Surface Construction Algorithm". *Computer Graphics*, vol. 21, no 4: 163-169

Lynch K (1960). *The Image Of The City*. MIT Press, Boston, MA

Lynn G (1999). *Animate Form.* Princeton Architectural Press, New York

Lopez MJ (2003). *Retail Store Planning And Design Manual*. ST Publications, Cincinnati, Ohio:169-170

Luchins A (1968). "Max Wertheimer"*.* In Sills DL (ed.), *The International Encyclopedia of the Social Sciences.* Macmillan and The Free Press, New York, vol. 16: 158-175

March, L: The Logic of Design and the Question of Value. In *The Architecture of Form*, March (ed.), Cambridge University Press, 1976a, pp 1–40

March, L: A Boolean description of a class of built forms. In *The Architecture of Form*, March (ed.), Cambridge University Press, 1976b, pp 41-73

March, L, Echenique, M and Dickens, P: Polemic for a Structural Revolution. In *Models of Environment*, Pridgeon, M (ed), Architectural Design, vol. 151, 1971

March, L. and Steadman, J. P.: *The Geometry of Environment – an introduction to spatial organization in design*. RIBA Publications, London, 1971

Markhede H and Miranda P: *Spatial Positioning Tool - a Prototype Software and Some Background Correlation Data*. In Proceedings of the 6th International Space Syntax Symposium, Istanbul, 2007: pp. 102.01 – 102.11

Martinetz T and Schulten K: *A Neural-Gas Network Learns Topologies*. In Kohonen, Maekisara, Simula and Kangas (eds.): *Artificial Neural Networks*, Elsevier Science, North-Holland, 1991

Maturana, H: *Biology of Cognition*. University of Illinois, 1970

Maturana H: *Biology of Language: The Epistemology of Reality*. In Miller and Lenneberg (eds): *Psychology and Biology of Language and Thought: Essays in Honor of Eric Lenneberg*. Academic Press, New York, 1978

Maturana, H: *Metadesign.* Instituto de Terapia Cognitiva INTECO, Santiago de Chile, 1997: http://www.inteco.cl/articulos/006/texto_ing.htm

Mayer R, Merkl D and Rauber A: *Mnemonic SOMs - Recognizable Shapes for Self-Organizing Maps*. In proceedings of the Fifth International Workshop on Self-Organizing Maps (WSOM), Paris, 2005

Mayurama M:*Second-order Cybernetics: Deviation-Amplifying Mutual Causal Processes.* American Scientist, vol 5, issue2, 1963: pp 164-179

McLuhan, M: Understanding Media – The Extension of Man. MIT Press, Cambridge MA, 1994

Metropolis N, Rosenbluth, AW, Rosenbluth, MN, Teller, AH, Teller E: *Equation of State Calculations by Fast Computing Machines*. The Journal of Chemical Physics 21 (6): 1087, 1953

Minsky M and Papert S: *Perceptrons – an introduction to Computational Geometry.* MIT Press, Cambridge MA, 1969

Miranda P (2007). "Out of Control: The media of architecture, cybernetics and design". In Lloyd Thomas K (ed.), *Material Matters – Architecture and Material Practice*, Routledge, Oxford

Miranda P and Coates P (2000). *Swarm Modelling - The use of Swarm Intelligence to generate architectural form.* In Proceedings of the Generative Arts conference, Milan

Miranda P and Derix C (2009). *Aedas R&D: Global Practices of Computational Design*. In proceedings of the 29th American Computer-Aided Design in Architecture conference (ACADIA), Chicago: 242-244

Miranda P (2014)." Programs as Paradigms". In Derix C and Izaki Å (eds.), *Empathic Space: The Computation of Human-Centric Architecture. Architectural Design*, vol. 85, issue 2, Wiley and Sons, London: 66-73

Mitchell W (1990). *Logic of Architecture*. MIT Press, Cambridge MA

Moller C (2005). "Planning Tools". In Samantha Hardingham (ed.), *The 1970s is Here and Now,* Architectural Design, vol. 75, issue 2, Wiley and Sons, London: 44-49

Montello DR (1991): *Spatial Orientation And The Angularity Of Urban Routes*. *Environment and Bahaviour,* vol. 23, issue 1: 47-69

Montello DR (2001). "Spatial Cognition". In Smelser NJ and Bates PB (eds.), *International Encyclopedia of the Social and Behavioural Sciences*, Pergamon Press, Oxford: 14771-14775

Muther R: *Systemic Layout Planning – An Overview*. http://www.hpcinc.com/rma/rma.asp: accessed August 2012

Negroponte N (1970). *The Architecture Machine – Toward a more Human Environment.* MIT Press, Cambridge MA

Nunes de Castro L (2006). *Fundamentals of Natural Computing: Basic Concepts, Algorithms and Applications.* Chapman & Hall/ CRC, Boca Raton

Oborn P (2013). *Al Bahr Towers - The Abu Dhabi Investment Council Headquarters*. Wiley, London

O'Rourke J (1994). *Computational Geometry in C.* Cambridge University Press, Cambridge

Otto F (2008). *Occupying and Connecting: Thoughts on Territories and Spheres of Influence with Particular Reference to Human Settlement*. Edition Axel Menges, Fellbach

Otto F and Rasch B (1995). *Finding Form: Towards an Architecture of the Minimal*. Edition Axel Menges, Fellbach

Panait LA and Luke S (2004). *Ant foraging revisited*. Submitted to the Ninth International Conference on the Simulation and Synthesis of Living Systems (ALIFE9)

Papert S (1980). *Mindstorms: Children, Computers, and Powerful Ideas*. Basic Books, New York

Pask G (1958). *Physical analogues to the growth of a concept*. Mechanization of Thought Processes, Symposium 10, National Physical Laboratory, H.M.S.O., London: 765- 794

Pask G (1969). "The Architectural Relevance of Cybernetics". In *Architectural Design*, Special Issue, vol. 9, Wiley and Sons, London

Pask G (1976): *Conversation Theory – applications in education and epistemology.* Elsevier, Amsterdam

Pearman H (2005)."Caruso St John/ Cover Versions". *Architectural Journal*, 6th October: 29-41

Petrovic I and Svetel I (1993). *Sketch of a Distributed Architectural Design System.* In proceedings of eCAADe, Eindhoven

Phillips D (2004). *Daylighting - Natural Daylight in Architecture*. Elsevier, Oxford

Piaget J and Inhelder B (1967).*The Child's Conception of Space.* FJ Langdon and JL Lunzer (trans.), WW Norton, New York

Pohl J and Myers L (1994). "A Distributed Cooperative Model for Architectural Design". In Carrara G and Kalay YE (eds.), *Knowledge-based Computer-Aided Architectural Design*. Elsevier Science, Amsterday: 205-242

Porta S, Strano E, Iacoviello V, Messora R, Latora V, Cardillo A, Wang F and Scellato S (2009): Street Centrality and densities of retail and services in Bologna, Italy. In *Perception*, vol. 38: 450-465
Price C (2003). *Cedric Price: The Square Book*. Wiley Academy, Chichester

Prim RC (1957): "Shortest connection networks And some generalizations". Bell System Technical Journal vol. 36, issue 6: 1389–1401

Rana S and Batty M (2004)."Visualising The Structure Of Architectural Open Spaces Based On Shape Analysis". *International Journal of Architectural Computing*, vol. 1, no 1: 123-132

Reynolds CW (1987): "Flocks, Herds, and Schools: A Distributed Behavioral Model". In *Computer Graphics*, vol. 21(4): 25-34

Reynolds CW (1999). *Steering Behaviors For Autonomous Characters*. In proceedings of Game Developers Conference, San Jose, CA. Miller Freeman Game Group, San Francisco, CA: 763-782

Resilient Infrastructure and Building Security (RIBS) (2013): *D4.4 Generic Infrastructure Report*. Koch D (ed.), project internal report

Richens P (1994). "Does Knowledge Really Help? CAD Research at the Martin Centre". In Carrara G and Kalay YE (eds.), *Knowledge-based Computer-Aided Architectural Design*. Elsevier, Amsterdam: 305 – 325

Rittel H and Webber M (1973). Dilemmas in a General Theory of Planning. *Policy Sciences*, vol. 4, Elsevier, Amsterdam: 155 – 169

Rogers R (ed.) (1999). *Towards An Urban Renaissance: Final Report Of The Urban Task Force*. E&FN Spon, London

Rosenblatt F (1957): *The Perceptron--a perceiving and recognizing automaton*. Cornell Aeronautical Laboratory, Report 85-460-1

Rosenblatt F (1958). "The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain". *Psychological Review*, vol. 65, no 6, Cornell Aeronautical Laboratory: 386–408

Rosenman M and Gero J (1993)."Creativity in Design Using a Design Prototype Approach". In Gero J and Maher ML (eds.),*Modelling Creativity and Knowledge-Based Creative Design*, Lawrence Erlbaum Publishers, Hillsdale NJ: 109 – 138

Rosenmann M, Gero J and Maher ML (1994): "Knowledge-Based Research". In Carrara G and Kalay YE (eds.), *Knowledge-Based Computer-Aided Architectural Design*, Elsevier Science, Amsterdam: 327-377

Rossi A (1966): *L'architettura della Citta*. Marsilio Editori, Padova

Rowe C (1983). "Program vs Paradigm"*. The Cornell Journal of Architecture*, vol. 2, Rizzoli, New York: 9-19

Rumelhart DE, McClelland JL and the PDP Research Group (1986). "Parallel Distributed Processing: Explorations in the Microstructure of Cognition". Foundations, vol.1, MIT Press, Cambridge MA

Rumelhart DE, Smolensky P, McClelland JL and Hinton GE (1986). "Schemata and sequential thought processes in PDP models". In McClelland JL and Rumelhart DE (eds.), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. Psychological and Biological Models vol. 2, MIT Press, Cambridge, MA:7-57

Sadler S (1998). *The Situationist City.* MIT Press, Cambridge MA

Sadler S (2005). Archigram - Architecture without Architecture. MIT Press, Cambridge, MA

Sailer K, Budgen A, Lonsdale N, Turner A and Penn A: "Pre and Post Occupancy Evaluations in Workplace Environments Theoretical Reflections and Practical Implications". *Journal of Space Syntax*, vol. 1, issue 1: 199-213

Schön D (1983). *The Reflective Practitioner: How Professionals Think in Action.* Basic Books, New York

Scotini M (2006). *Gianni Colombo – The Space Device.* Exhibition catalogue text, Skira, Milano: http://www.archiviogiannicolombo.com/de/Bibliographie.htm, accessed 29.04.2015

Schmarsow A (1894). *Das Wesen der architektonischen Schoepfung.* Inaugural lecture, University of Leipzig, 8 November 1893, published in Karl W Hiersemann, Leipzig: 10–23

Schwarzer MW (1991).”The Emergence of Architectural Space: August Schmarsow’s Theory of Raumgestaltung”. *Assemblage,* vol. 15: 48–61

Shea K, Sedgwick A and Antonuntto G (2006). “Multicriteria Optimization of Paneled Building Envelops Using Ant Colony Optimization”. In Proceedings of Intelligent Computing in Engineering and Architecture, *Lecture Notes in Artificial Intelligence*, Springer, Heidelberg: 627-636

Simon H (1969). *The Sciences of the Artificial.* MIT Press, Cambridge MA (third edition 1996)

Simon H and Newell A (1958). “Heuristic Problem Solving: The Next Advance in Operations Research”. In *Operations Research*, vol. 6, no. 1

Srinivas N and Deb K (1995). “Multi-Objective function optimization using non-dominated sorting genetic algorithm” . In *Evolutionary Computation*, vol. 2, issue 3: 221-248

Steadman P (1970): “The automatic generation of minimum standard house plans”. LUBFS working paper no. 23, Cambridge University Press; published in *Models of Environment,* Pridgeon M (ed), Architectural Design, vol. 151, 1971: 320-321

Steadman P (1979). The Evolution of Designs: *Biological Analogy in Architecture and Applied Arts*. Routledge, London

Steadman P (1983). *Architectural Morphology: An Introduction to the Geometry of Building Plans*. Pion, London

Steadman P (1998). Sketch for an archetypal building. *Environment and Planning B: Planning and Design,* 25th Anniversary Issue: 92-105

Steadman P and Mitchell L (2010). Architectural Morphospace: mapping worlds of built forms. *Environment and Planning B: Planning and Design*, vol. 37: 197-220

Steadman P (2014). *Building Types and Built Forms*. Troubadour, Leicester

Stiny G and Gips J (1972). *Shape Grammars and the Generative Specification of Painting and Sculpture*. In proceedings of IFIP Congress 71, Amsterdam:1460-1465

Stonor T (2014): *Predictive Analytics for Urban Planning, Building Design and Spatial Economics.* Presentation at Future Cities workshop, Singapore, 11th march 2014: http://de.slideshare.net/tstonor/tim-stonor-predictive-analytics-using-space-syntax-technology, accessed 05.15.2015

Tabor P (1971). “Structures of Organization”. In *Models of Environment*, Pridgeon M (ed.), Architectural Design, vol. 151: 315 - 19

Theraulaz G and Bonabeau E (1995): “Modeling the collective building of complex architectures in social insects with lattice swarms”. *Journal of Theoretical Biology* , vol. 177: 381-400

Theraulaz G (2014). “Embracing the Creativity of Stigmergy in Social Insects”. In Derix C and Izaki Å (eds), *Empathic Space – the Computational of Human-centric Architecture.* Architectural Design, vol. 85, issue 2, Wiley and Sons, London: 54-59

Theraulaz G and Bonabeau E (1999). „A Brief History Of Stigmergy”. *Artificial Life* vol. 5: pp. 97-116

Transport for London: *Measuring Public Transport Accessibility Levels – Summary.* http://data.london.gov.uk/documents/PTAL-methodology.pdf, accessed 01.09.2014

Tufte E (2006). *Beautiful Evidence*. Graphics Press, Cheshire CT

Turner A (2000). *Angular analysis: a method for the quantification of space*. Working Paper 23, Centre for Advanced Spatial Analysis, UCL, London

Turner A (2001). *Depthmap – a program to perform visibility graph analysis.* In proceedings of 3rd Space Syntax Symposium, Atlanta: 31.1-9

Turner A (2006). *An ecomorphic theatre as a case study for embodied design*. In proceedings of the Generative Arts conference, Milan

Turner A (2009). “The Role of Angularity in Route Choice An Analysis of Motorcycle Courier GPS Traces”. In Proceedings of COSIT 2009, *Lecture Notes in Computer Science*, Springer, Berlin/Heidelberg

Turner A., Doxa M, O’ Sullivan D and Penn A (2001). *From isovists to visibility graphs: a methodology for the analysis of architectural space. Environment and Planning B*, vol. 28, issue 1: 103-121

Tschumi B (1994). *Manhattan Transcripts.* Academy Editions, London

Tversky, B (2000). “Remembering Space”*.* In Tulving E and Craik FIM (eds.), *Handbook of Memory*, Oxford University Press, New York: 363 - 378

Qu H, Zhou H and Wu Y (2006).*Controllable and Progressive Edge Clustering for Large Networks.* In proceedings of the 14th International Conference on Graph Drawing, Karlsruhe: 399–404

UrbanBUZZ (2010). *The Complete UrbanBUZZ*. UCL Press, London: 184-189

US General Services Administration (2006). *Workplace Matters*. Public Buildings Services, Washington DC: www.gsa.gov/workplace accessed June 2012

Van Tonder G and Ejima Y (2002). "Flexible Computation of Shape Symmetries within the Maximal Disk Paradigm". *Man & Cybernetics,* IEEE Transactions on Systems

Van Tonder G (2004). *Fractal potential of 'emptiness' in Japanese dry rock gardens*. In proceedings of FFRACTARQ'04, Madrid

Vischer JC (2007). "The Concept of Workplace Performance and its Value to Managers". *California Management Review*, vol. 49, issue 2: 62-79

Von Foerster H (1984). *Observing Systems*. Intersystems Publications, Seaside CA: 288-309

Walter WG (1950). "An imitation of life". In *Scientific American*, vol. 182, issue 5: 42–45

Wasserman S and Faust K (1994). *Social Networks Analysis*. Cambridge University Press, Cambridge

Weiler K and Atherton P (1977). "Hidden Surface Removal using Polygon Area Sorting". In *Computer Graphics*, vol. 11, issue 2: 214-222

Wertheimer M (1938). "Laws of Organization in Perceptual Forms". In Ellis W (ed.), *A source book of Gestalt psychology.* Routledge & Kegan Pau, London: 71-88

Whyte W (1980). *The Social Life of Small Urban Spaces.* Project for Public Spaces publisher, New York

Willoughby T (1971). "Evaluation Circulation Performance". In Pridgeon M (ed), *Models of Environment*, Architectural Design, vol. 151: 314-315

Yaneva A (2012). *Mapping Controversies in Architecture.* Ashgate, Farnham

Zhu P and Wilson RC (2005). *A Study of Graph Spectra for Comparing Graphs*. In proceedings of the British Machine Vision Conference: 679–688

Zentrum fuer Kunst und Medientechnologie (ZKM) (2004). *Die Algorithmische Revolution.* Exhibition brochure, Engelhardt & Bauer, Karlsruhe

11 GLOSSARY OF ABBREVIATIONS (ALPHABETICAL)

ACO	ant colony optimization: an algorithm to simulate ant colony dynamics as a graph introduced by Marco Dorigo (Colonori et al 1991)
AI	artificial intelligence
AL	artificial life
ANN	artificial neural network: an artificial intelligence meta-heuristics to simulate signal processing in of the animal brain in order to classify patterns (aka machine learning)
ANT	actor-network theory, proposed by Bruno Latour (1987)
API	application programming interface, allowing computer programmers to access a model code
BCO	British Council for Offices
BFS	breadth-first search: a computational graph search technique
BIM	building information modelling
CA	cellular automaton: an artificial intelligence meta-heuristic to generated patterns introduced by Stanislav Ulam (Langton 1995)
CABE	Commission for Architecture and the Built Environment, London
CAD	computer-aided design
CECA	Center for Evolutionary Computing in Architecture, founded by Paul Coates and the author at University of East London, from 2002 to 2010
CDR	Computational Design Research group, founded by author at Aedas architects, London, 2004-2014
DFS	depth-first search: a computational graph search technique
DIAP	Dipartimento di Architettura e Pianificazione (department of architecture and planning), Politecnico di Milano
FOV	field of view: used by computer agents in this context with a default setting of 120°
FuCon	Future of Construction: case study project, commissioned by the Fraunhofer Institute and funded by the German ministry of Infrastructure
GA	genetic algorithm: an artificial life meta-heuristic to simulate the search mechanism of natural selection (one algorithm in the family of Evolutionary Algorithms) introduced by David Goldberg (1989)
GNG	growing neural gas: an adaptive topology heuristic for artificial neural networks (ANN) introduced by Bernd Fritkze (1995)
GUI	graphical user interface
IAA	Integrated Associative Analysis: case study project in 6.3.4 as part of the RIBS project, based on a SOM classification
IAO	Fraunhofer Institute for Workplace Organization, Stuttgart
IDE	integrated development environment: a software development environment for specific programming languages to develop, debug and build code
KPI	key performance indicator
LUBFS	Centre for Land Use and Built Form Studies at Cambridge University
MSc	university degree of master of science
MST	minimum spanning tree: a sub-graph from a set of nodes that connects all nodes of a set with the shortest distance (Prim 1957)

NSMM	National September 11th Memorial Museum, New York
OD	origin-destination pair: usually for route calculations on graphs, a starting point is called the origin and the terminating point is called destination
OFSS	Open Framework for Spatial Simulation: the computational framework that integrates the presented case studies, constituting the operational output from 10 years R&D at CDR. It represents one research objective of this dissertation (8.3.3)
PTAL	public transport access levels: a methodology for mapping accessibility to public transport and indicator for development density used by urban planners
RIBA	Royal Institute of British Architects
RIBS	Resilient Infrastructure and Building Security: multi-model case studies summarized in 8.3.3 that was conducted within a European FP7 research collaboration
SOM	self-organizing feature map: a non-supervised self-organizing ANN introduced by Teuvo Kohonen (1981)
SOS	Self-Organizing Space: case study project in 6.3.2, based on the SOM algorithm
STG	Spatial Topology Graph: case study project in 6.2.4 based on the computational shape analysis technique called medial axis transform (MAT)
SynUrb	Synergetic Urbanism: case study project in 8.1.1
TfL	Transport for London
TU	Technical University
UCL	University College London
UEL	University of East London
USOM	User-centric Spatial Operations Model: abbreviation for the overall research objective to develop an open computational framework for human-centric spatial design
VPTA	visible polygon traversal algorithm: algorithm developed by Åsmund Izaki (Izaki and Derix 2013) in 6.1.2 for visibility analysis
VGA	visibility graph analysis: discretized syntactic analysis of a view shed introduced by Alasdair Turner (Turner et al 2001)
VR	virtual reality

CURRICULUM VITAE

PERSON

	Christian Wilhelm Derix
1973	born March 3rd, Frankfurt/Main, Germany

EDUCATION

1990	High School Diploma, Willits, CA, USA
1993	Abitur, Wilhelm Dilthey Gymnasium, Wiesbaden, Germany
1999	Bachelor of Science in Architecture (BSc), University of East London (UEL), London, UK
2001	Diploma of Architecture (DiplArch), UEL, London, UK
2001	Masters of Science (MSc), Computing and Design, UEL, London, UK

ACADEMIC

2001-9	Senior Lecturer, MSc Computing and Design, UEL
2004	Assistant Professor, Department of Architecture and Planning, Politecnico di Milano, Italy
2005-8	Assistant Professor, Institute for Architecture and Design, Technical University Vienna, Austria
2007	Senior Lecturer, MSc Adaptive Architecture & Computing, University College London (UCL), UK
2009-	Diploma Theses supervisor, Bartlett School of Architecture, UCL
2011-12	Visiting Professor, Chair for Emerging Technologies, Technical University Munich, Germany
2013-14	Visiting Professor, School of Architecture, University of Sheffield, UK
2013-15	Associate Professor, Masters in Architectural Design and Management, IE University, Madrid, Spain

PROFESSIONAL

1994	Assistant, Quagliata Glass Studio, Oakland, CA, USA
1994-5	Assistant, Stained Glass Restoration, Studio Forme, Rome, Italy
2001-2	Architectural Assistant, Stephen Rich architects, London, UK
2004-14	Director, Computational Design Research (CDR), Aedas architects, London, UK
2015-	Director, SUPERSPACE Design Research, Woods Bagot architects, Sydney, Australia