A Workflow for Efficient Data Collection and Simulation in Hospital Environments

Jelena Đorđić

Master's Thesis

2017

Master's Thesis

A Workflow for Efficient Data Collection and Simulation in Hospital Environments

Submitted in partial fulfillment of the requirements of the academic degree

Master of Science in Architecture MSc

Supervisor

Privatdozent Dipl. Ing. Dr. Gabrial Wurzer

259.1 Institute of Architectural Sciences, Vienna University of Technology

Digital Architecture and Planning

Author

Jelena Đorđić

Registration number 1229144

Vienna University of Technology, 2017

Abstract

Simulation is very useful tool in a planning process. Architects can use it to present their design but also to test them through various software and methods. Hospital building, as a concise combination of form and function, arises from a set of recommended pre-requisites that have a strong foothold in organizational processes and functional flows. If an architect can work and think not only trough drawing and rendering but also trough other media, such as simulation that employs time and user as variables, he achieve much better control over the project. Simulation can also be used for scientific purposes but the problem usually arises from the lack of data necessary for the simulation. Only architecture office that is involved in a project has the building plans in possession. No matter is it a completely new building or a renovation, current i.e. old plans one cannot obtain if he works outside of the responsible office.

For simulation, one needs to obtain usable and complete building plans, which requires permission from owner and further request at authority in charge. The process may take some time. When one gets permission to access the archives, he can find the plans most probably in paper form. Scanning the drawings with handheld or regular scanner is also a slow process and the digitized plans need to be adjusted. Since one needs only a few information from the plans for the simulation and not everything depicted with drawings, extracting and editing necessary information is inevitable.

In this work, we tried to improve current workflow, especially the process of obtaining data and their preparation for simulation.

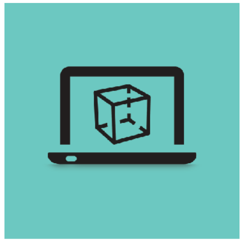
Table of Contents

Dedication

I dedicate this work to my family and my mentor, who had enough patient and kindness to support me along the way.

1.

Introduction



2.

Computer Simulation
as a Planning Tool for
Architect

3.

Related Work

4.

Improved Workflow

5.

Case Study Baden

6.

Discussion

1. Introduction

Computer simulation methods have been identified as valuable planning tools for guaranteeing cost efficiency, efficient workflows, reduced waiting and admission times.[1]Modelling our physical world and behaviour patterns within it offers an insight into how complex systems work. The goal is to not only to provide information on a system's performance but also to detect potential problems arising from its structure. As a planning tool, simulation can be used to collect and visualize data related to hospital planning either at the planning stage or later during the utilisation phase of a building where it is being used as means of analysis.

Planning a hospital is one of the most complex tasks in architecture. Although there are types of buildings that are structurally, much more complex - such as stadiums or industrial halls - hospitals stand out for their complexity in functional terms.

When planning, it is necessary to consider factors that directly affect the structural and spatial concept of a building. In addition to these internal factors, hospital planning also has to consider a multitude of external constraints, such as the social and economic profile of the community, local building regulations, the health profile of a region, topography and local climate. From these predominantly numerical and descriptive data, the physical structure of the health institution is created.

Planning is based on a space allocation plan stating which spaces serving what functions are needed for operation. After having elaborated a (seemingly) suitable plan, it would be very useful for architects and other types of planners if they could perform testing through simulation. In this way, it would be possible to determine e.g. the most crowded areas and so forth, in order to dimension spaces more adequately.

1 See e.g. the Winter Simulation Conference (http://informs-sim.org/), the largest existing simulation conference to date, which has an own track for healthcare applications (with sub-tracks such as capacity planning, simulation of emergency departments and patient scheduling).

One further advantage of simulation lies in the possibility to virtually test a design in consideration of expected flows (e.g. patient flow, flow of materials and so forth), so as to resolve eventual junctions (paths where clean and soiled materials cross), analyse lengths of the trajectories as well as their duration, so that movement within the building can be carefully designed as well.

However, such simulation studies are typically conducted far too late within the planning to be able to influence the design on a fundamental level. In extreme cases, simulation is made only after the building has been completed, which reduces its role to a means of analysis that facilitates future planning. However, this is a pity since it could be used as design tool in its own right - which is one of the goals that this thesis tries to achieve. Most hospitals are not build from scratch but refurbished. In order to carry out a credible simulation in that setting, actual hospital plans are necessary. Sometimes plans can be found in books and magazines but their usability depends on the drawing scale and an amount of information that resources contain (Figure 1a). Otherwise, it is very hard to obtain them, though: Architectural bureaus (Figure 1b) do not provide information easily, while the building inspection demands special permissions to obtain that data (Figure 1c).

Many countries have laws regulating access to public documentation (in the U.S., there is the Freedom of Information Act, while the Informationsfreiheitsgesetz has a similar function in Germany; regulations can even vary county-wise: in Vienna, it is possible to request to take a look into plans of private or public facilities at the Magistrate 37, while example). In all cases, the regulations authority reserves the right not to provide access to

Figure 1 Data inquiry a) Public library b) Architecture office c) Municipal building inspection

information that would possibly jeopardize public safety.

When it comes to private facilities, the permit is issued only based on the authorization by the owner. Public property plans can also be viewed for scientific purposes, but the consent of the owner is still inevitable. When the magistrate gives access to the archive based on the permit issued, it is necessary to scan or copy the plans on the spot. Digitalization i.e. scanning of large formats is also possible. In Vienna, this function is performed by the Magistratsabteilung 21 Stadtteilplanung und Flächennutzung) with a specific statutory fee. In case of historic buildings, it is somewhat easier to obtain the plans, since the stated aspect of public safety is generally not an issue.

However, the quality of these plans are largely inadequate (old blueprints, decayed paper, fragments missing), so that it is often not an option to obtain a historical base plan and to add adaptations to that. Another shortcoming of historical documentation lies in its availability: city libraries often dispose planning documentation to make space in their archives. Building plans can also be found in various books or publications, which would seem a good source even for present-day architecture. However, in this case the plans are mostly incomplete and only show segments of work that are most suitable for presentation. An additional difficulty is that drawings in books are very small and there is a loss in quality due to the printing process, making it difficult to vectorise the image. The result might also be difficult to scale properly because of lacking map scales.

It is possible to imagine a scenario in the future where hospital plans will be published on web mapping services such as Google Maps.[2] For now, mainly shopping malls, train stations, airports, universities or museums can be found there.[3]

Hand-drawn plans are even more problematic than published digital plans: Different (non-standardised) symbols may be present, and the plans might already contain inaccuracies due to manual drawing. Summarizing, it is fair to say that there is a large domain of problems that need to be tackled after digitisation. How much time and effort must be spent on subsequent processing of

2 http://maps.google.com
3 own survey of floor plans in Google Maps conducted in June 2017

plans ultimately depends on the inherent quality of the drawings, the quality of scan and what details need to be captured as input for subsequent simulation.

The aim of my diploma project is to simplify the process of obtaining digital floor plans, while at the same time providing some quality improvements to the overall process. Instead of tediously gathering plans through the local regulations office (requiring permission of the building owner), photographs of fire escape plans will serve as basis for my approach. However, it is always possible to apply the presented methods also to richer data representation (scanned books, exported digital drawings), if necessary.

(Work structure) The goal of my work is to find building plans and extract/adjust necessary data in a more simple way in comparison with the current workflow.

Chapter 1 introduces the problem statement. The current process of obtaining and preparing building plans for simulation is relatively slow and complicated because it requires a lot of energy and time in order to find appropriate data and also significant manual labor concerning those for simulation.

Chapter 2 discusses the work environment. Here will be shown which software packages can be used in the process, what kind of input data are we seek for and possible methods how we can reach those data.

The current State of Art is given in the **Chapter 3**. It is divided in two sections: Early-Stage Hospital Planning and Recognition of Floor Plans. The first section concerns the very simulation process that derives from various approaches and methods. The second section offers an insight in some previous works related to recognition i.e. digitizing building plans.

A possible answer on the stated problem is suggested in the **Chapter 4**. It concentrates to the main step stones in the current workflow: obtaining building plans, their adjustment and vectorization. Suggested solutions take into consideration already used methods and previous knowledge and offers an improvement that saves time and reduces manual labour.

Theoretically derived solution is tested trough Case Study in the **Chapter 5**. Discussion and final review are given in the **Chapter 6**. Figure 2 illustrates the aforementioned work structure.

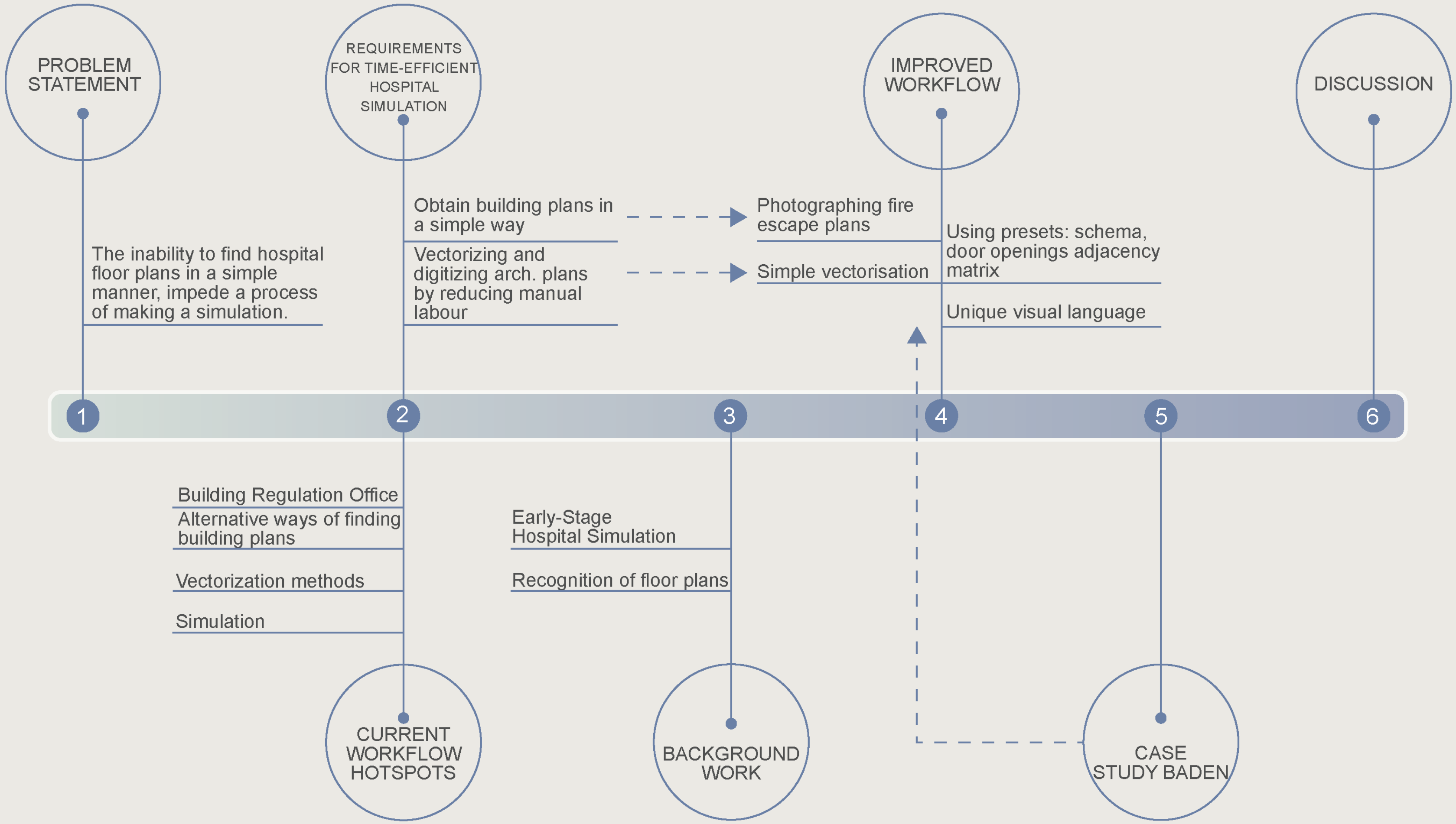


Figure 2 Structure of the Master's Thesis "A workflow for efficient data collection and simulation of hospital environment"

1.

Introduction

2.

Computer Simulation
as a Planning Tool for
Architect

3.

Related Work

4.

Improved Workflow

5.

Case Study Baden

6.

Discussion

2. Computer Simulation as a Planning Tool for Architects

It is difficult to imagine a building purely based on two-dimensional floor plans. Over the centuries, architects have been concerned with three-dimensional representations such as perspective/isometric drawings or hand-sketches. The goal of all of these richer representations is to provide information on structural relations and spatial arraignments such that comprehension of aesthetics and function of architectural design requires less effort.

Methods for depicting buildings by means of perspective drawings date back to fifth century BC and the art of ancient Greece.[4] Oblique perspective, used by Chinese artist between first and eighteenth century comes most probably from ancient Rome. Some of the paintings found in the ruins of Pompeii show a remarkable realism and perspective for their time.[5] However, it took until the renaissance until the underlying laws of perspective, which are still in use today, were formally described.[6] Renaissance artists such as Brunelleschi and Dürer in their own works on linear perspective used Euclid´s Optics written around 300 BC.[7]

In last few decades, digital techniques have largely replaced hand-drawings. In addition, the workflow has changed significantly: Renderings are prepared in parallel to the development of floor plans, in order to address questions related to aesthetic aspects of a building. This is not enough, though, to understand fully a building's operation: Aspects of work routines, energy and so forth require a fourth dimension - time.

4 Performance in Greek and Roman Theatre, George W.M.Harrison, Vayos Liapis, 2013, Koninklijke Brill NV, Leiden, ISBN: 0169-8958, p.1203 own survey of floor plans in Google Maps conducted in June 2017

5 Pompeii: Its Life and Art, August Mau, 2014, Literary Licensing, LLC, 2014, ISBN: 1497842697, p. 457

6 Britannica, Linear perspective, https://www.britannica.com/art/linear-perspective, accessed 16.07.2017

7 Catching the Light: The Entwined History of Light and Mind, Arthur Zajonc, 1993, Oxford University Press, ISBN: 0553089854, p.25

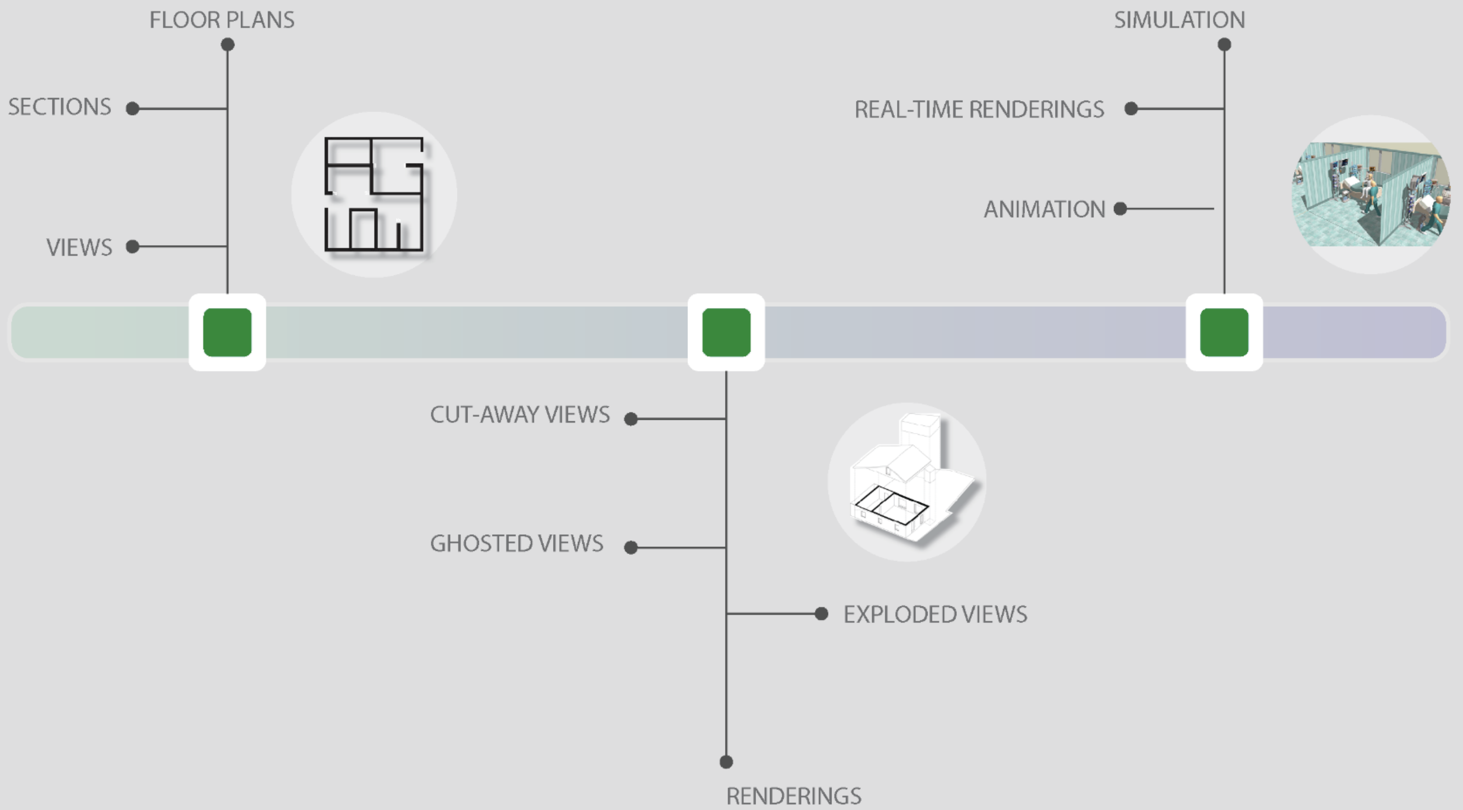


Figure 3 Mediums of architectural expression

Simulation is built on this fourth dimension: Based on a spatial context (two- or three-dimensional plans coming from Computer-Aided Design [CAD], Building Information Modelling [BIM] software or Geographical Information Models [GIS]), it computes flows over time (e.g. physical flows used for heating, ventilation and air conditioning [HVAC]; flows of building users and material) that can inform a design on dynamic aspects that are crucial for operation. Figure 3 depicts current tools used by architects in order to obtain images of a building before or during construction process.

However, if simulation is unreasonably time-consuming and requires extensive resources, its benefit for the design process is debatable. Therefore, it is imperative to simplify both the act of data collection as well as the model used. In this manner, time and resources can be devoted to an analysis of certain problem and not on the simulation itself.

2.1. Simulation Toolkit

> The Roman architect Vitruvius in his treatise De Architectura (on architecture) defines the principles of good architecture as firmatis (durability), utilitas (utility) and venustatis (beauty).[8] If architecture is characterized by these criteria, can the same be said about hospital planning - or is this field rather a mathematical than architectural one - since the design space is severely restricted by e.g. functional and structural demands, as well as questions of operation.

Indeed, the way we approach hospital planning has not much in common with design: Instead of simultaneous development of a layout, structure and design, priority is given to space allocation, using only the strict logic of function. Given that (1.) a building's Gross Floor Area (GFA), Net Internal Area (NIA) and space allocation plan are usually pre-defined and (2.) medical equipment imposes a certain room layout, design-

8 The ten books on architecture, Vitruvius, translated by Morris Hicky Morgan, 1914, Harvard University Press, p.31

driven manipulations are quite moderate when comparing e.g. to residential buildings. This does not mean that building design is completely neglected, but strongly influenced by rigid utilitarian requirements. If we take in consideration the average size of health facilities, the functional character of hospital architecture becomes even more understandable.

Louis Sullivan's modernist mantra 'form (ever) follows function'[9] is still the standard approach to hospital architecture: Form is derived from a set of demands, e.g. traffic, economic setting, sanitary standards and functional requirements. Similar to industrial buildings, where the main task is to provide enclosed space large enough to shelter machines and workers, hospital architecture focuses on the workflow of its building users. Even in regards to design, functionalist guidelines are being followed: Materials, colours and furniture design, for example, are all under the influence of this doctrine. In that same mindset, we find that even a hospital's layout can be a result of thoughts on efficiency and capacity conducted on a strategic level.[10] Elements of a design - such as the circulation - are determined by the intersection of multidisciplinary pathways of care. This can happen not only in physical reality (i.e. space and time) but also in the more abstract information space of care giving (e.g. through dependencies between disciplines). In order to analyse these intersections while planning layout, we have developed a computer simulation that deals with the only the first aspect - intersections in space and time.

Various efforts have been made in past years (see also section 3 [related work]): By using diverse scenarios, planners are able to analyse total travelling time for patients and personal, patient waiting times, occupancy and so forth. The focus

9 Britannica, Louis Sullivan, https://www.britannica.com/biography/Louis-Sullivan, accessed 16.07.2017

10 A framework for health care planning and control, Hans Erwin W., Houdenhoven, Mark and Hulshof, Peter J.H., 2011, Memorandum 1938, Department of Applied Mathematics, University of Twente, Enschede. ISSN 1874-4850, p. 4, http://doc.utwente.nl/76144/1/memo1938.pdf, accessed 5 Mai 2017

in all of these analysis tasks lies on capacity planning - i.e. downsizing resources or increasing their utilization, elaborating an appropriate scheduling scheme(more appointments, less walk-in patients), at the same time reducing waiting times for patients.

Some of these methods have been implemented into tools; however, these are mostly experimental in a proof-of-concept stage. Some examples of commercial implementations being used in practice are:

FlexSim.[11] FlexSim is a user-friendly software that operates in a 3d environment with drag-and-drop controls and intuitive possibilities for patient-centred design (especially in a specialised version called FlexSim HC - FlexSim for Healthcare). The software uses a discrete-event [DES] approach (serving units with queues), although one could also say that FlexSim is agent-based [ABS] (patients and care-givers are visualized interactively).

11 William B. Nordgren. 2002. Flexsim: Flexsim simulation environment. In Proceedings of the 34th conference on Winter Simulation: exploring new frontiers (WSC '02). Winter Simulation Conference 250-252.

Simio.[12] This software package provides a true object-based 3D modelling environment without need for programming. Models are built using an object-based approach, with different library elements for different fields (e.g. Simio-HC - Simio for Healthcare).

The modelling paradigm of Simio is similar to Petri Nets[13]: Each object may carry out a set of processes. Processes are defined as sequence of activities. Each process instance transitions through that activity sequence using a process token, which also carries the state. Simio also has an (animated) 3D visualisation of objects undergoing processes, so the approach would perhaps also qualify as an agent-based simulation model.

Simul8. Simul8 is a software focusing especially on discrete-event simulation (servers and queues). In contrast to FlexSim and Simio, it is purely 2D-based and does only involve animation on a very crude level. The used metaphor is a library of elements which are interconnected to form a discrete-event network, e.g. a queue and a server.

12 C. Dennis Pegden and David T. Sturrock. 2009. Introduction to Simio. In Winter Simulation Conference (WSC '09). Winter Simulation Conference 314-321.
13 James L. Peterson: Petri Net Theory and the Modeling of Systems. Prentice-Hall, Englewood Cliffs NJ 1981, ISBN 0-13-661983-5.

SimCad. This software is used for simulating process-based environments. It is intended not only for the healthcare domain (SimCad Pro Health) but also for logistics, manufacturing and warehouse operation among many other fields.

The modelling metaphor is not unlike that of FlexSim, in that servers and queues are defined in a 3D environment, which is simulated and animated. As result, one can perform e.g. a length of stay analysis, optimize patient flow and perform supply chain analysis.

AnyLogic. Anylogic is a software that offers a multi-model metaphor (one can model using discrete-event, agent-based, system dynamics [SD]). The interface is generally two-dimensional, even though there is the possibility to have the simulation be visualized and animated in 3D in a post-step. Customizing the software is possible by programming. However, one generally does not need to be a programmer to be able to develop a simulation model (components are available via drag-and-drop).

Other packages. Some commonly used 3D software packages for architecture such as Maya or 3dsMax are intended for animation purposes rather than professional simulation analysis.

However, they could also be used for simulation development (scripting a custom simulation as means) - which would be very useful in designing process since visual results are significantly better (through rendering) what simulation packages provide.

Additionally, there is quite a wide range of agent based modelling toolkits requiring different levels of programming knowledge - from no programming to professional programmer level. One of these platforms is **Netlogo**,[14] which is an agent-based programming language and integrated (2D and 3D) modelling environment. In contrast to proprietary healthcare simulation software NetLogo is an open-source software that comes with an extensive models library and documentation, which is why I have chosen to use it for the development of this thesis.

14 Wilensky, U. 1999. NetLogo. http://ccl.northwestern.edu/netlogo/. Center for Connected Learning and Computer-Based Modeling, Northwestern University. Evanston, IL.

2.2. Input Data

As already mentioned in the previous chapter, a simulation requires appropriate input data.

Architectural plans can be found in either a paper or digital form.

- Paper form is inappropriate for explicit computer use since the drawings need to be digitized. The process can be conducted through scanning or photographing of architectural plans.

- Digital form is nowadays the standard, however, architecture bureaus do generally not provide this data because of intellectual property / copyright issues, and only submit what is required by the regulatory process (currently printouts).[15] Thus, this thesis still targets digitizing the paper form.

15 This might change with the adaptation of Building Information Models (BIM): In the UK, architecture offices are required to submit a BIM model for the public sector.

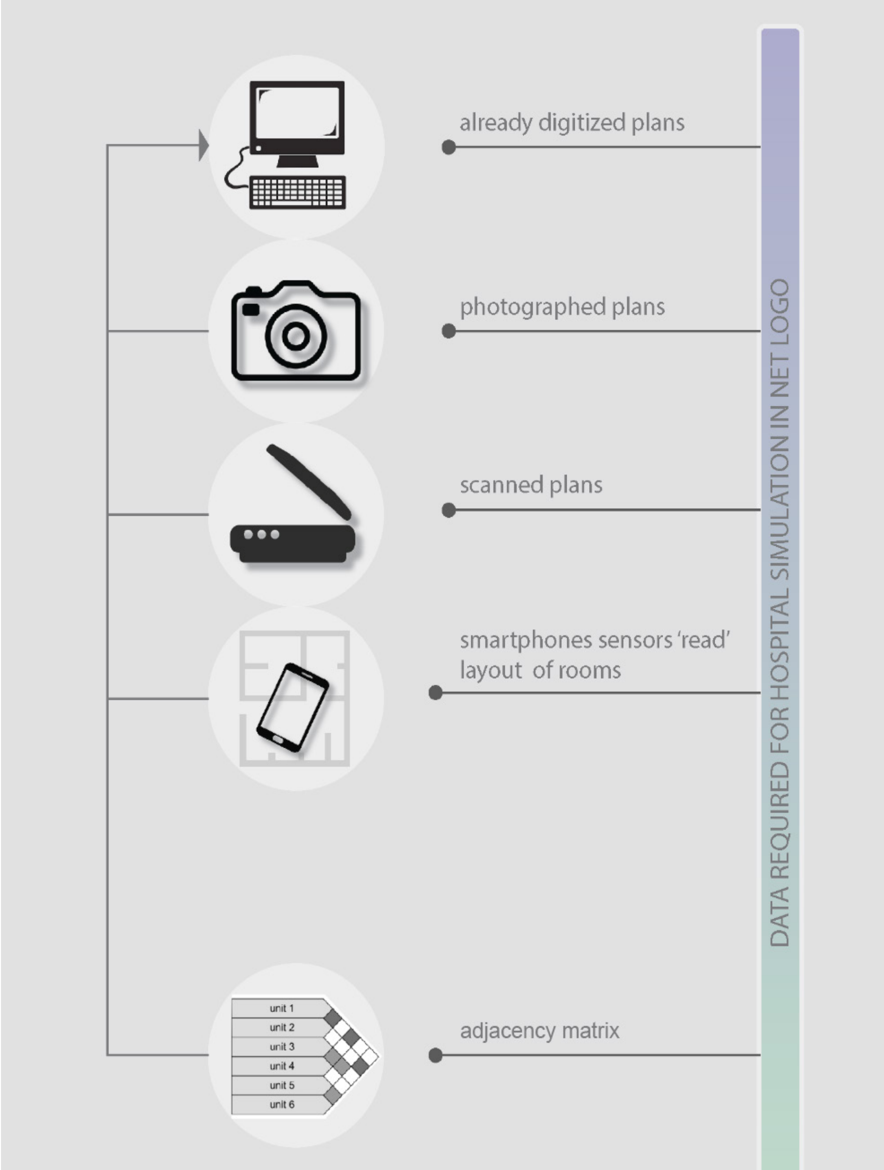


Figure 4 Data required for hospital simulation in NetLogo

In general, building plans should be submitted to authorities at the scale 1:100. Considering such large paper sizes, photographing i.e. scanning is very challenging. The plans are commonly archived in Magistrat 37 (referring to the city of Vienna). After all mandatory permissions had been issued, it is possible to scan the plans on the spot (no borrowing of the plans for professional scanning is usually possible).

The best option in that case is to use a handheld scanner, however this is prohibitively slow for a large amount of plans (photography is employed in that case). The scanned or photographed drawings must then be adjusted by cropping, scaling, skewing and combining multiple images (also see section 2.4 for details).

A simulation workflow is initiated with either scanned or photographed plans. Photographing within an existing building is an alternative to the previous approach: If we manage to find convenient floor plans *outside the Magistrate* and *without having to wait for permission*, we can use any technical means beginning with smartphones and ending with large scanners, in order to obtain digitized data. Fire evacuation plans are an excellent choice in that setting, since they must be available (fire regulation laws) and usually show the most recent plans. The photographed plans must still be prepared for vectorization. Photoshop, Gimp or any equivalent photo editing software is suitable for this task. In Chapter 4 we discuss details of the latter workflow, which is the method used throughout this thesis.

Another technique of providing building plans would be surveying (i.e. photogrammetric reconstruction, laser scanning and so forth; see 2.3 for a detailed description). Manifold attempt has been already made in order to construct floor plans by using smartphones; however, this is a tedious process if we consider that we would need an access to every room in a hospital building.

Adjacency matrices sometimes also serve as necessarily input for simulations. They give relations between hospital departments regarding intensity of personal/patient flow . Information on patient flow could be obtained from hospital information system, but if we do not have an access to the system, there is no possibility to establish precise frequencies based on current online data. We can access statistic reports

2.3. Indoor Mapping as Sources of Building

on admitted patient diagnosis on the state level, but we cannot derive any conclusion on patient flow.[16] For instance, the exact number of admitted patients with heart conditions does not provide information which departments such patient visits.

We can assume that he/she will be treated in the cardiology department, but we cannot tell if he/she will also visit the laboratory or pneumatology department based only on the fact that he/she has a heart condition.

The information that we can gather online are not suitable for comparing or combining. For this reason, adjacency matrices stating intuitive frequencies between departments are established by experts or with reference to literature on hospital planning.[17]

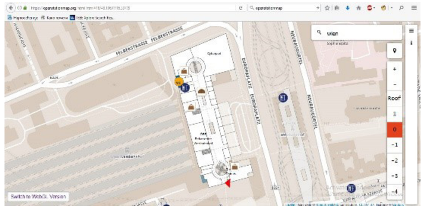
16 W Lorenz, M. Bicher, G. Wurzer: “Adjacency in hospital planning - Using Newton´s differential equation”; Proceedings of MathMod 2015, IFAC Online Conference Paper Archive, (2015), 6 pages.
17 cf. e.g. Neufert, E and Neufert, P (2012), Architect’s Data (4th Edition), Wiley-Blackwell.

Services such as Google Maps provide outdoor spatial information. On the other hand, there is no service that offers a general insight into indoor environments. Google maps offers an option for viewing/editing indoor maps as well, but the number of created locations is still very low.

Although indoor mapping of hospitals would be very useful not only for architects but for visitors as well, there is no hospitals yet to be found on the list of mapped buildings.[18]

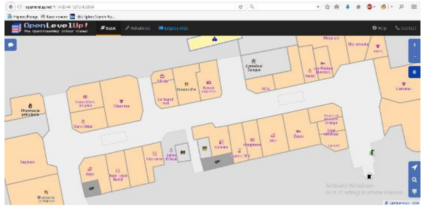
OpenStreetMap (OSM), an outstanding example of volunteered geographic information (VGI) aims creating a free and editable map of the world. Several tagging schemes have been already proposed concerning indoor mapping but creators still need to overcome many challenges.

18 as per June 2017, see: https://support.google.com/maps/answer/1685827?hl=en



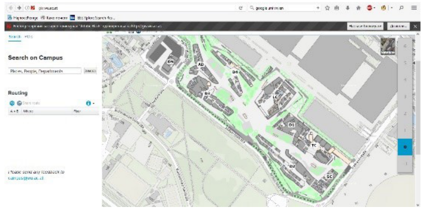
OpenStationMap is a project to map railway stations with routing, indoor, 3D and of course station specific features.

Westbahnhof, Wien

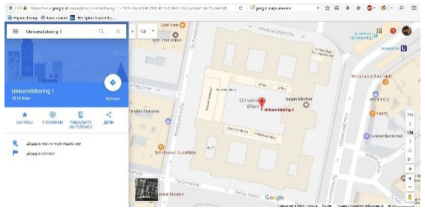


Open LevelUp is an interactive web map which displays the inside of buildings, split in levels.

Le Coudoulet, Orange



Vienna University of Economics and Business Administration Indoor routing and maps.
The WU Vienna has indoor maps and navigation.



One of a few buildings from Vienna added to **Google Indoor Maps.**

University of Vienna

Figure 5 Indoor-mapping projects

However, there are already independent attempts to depict information on spatial organization of buildings (service OpenStationMap for example is successful project to map railway stations, including indoor mapping). There is, however, still no hospital-mapping sub-project to the best of our knowledge.[19]

Figure 5 shows projects related to OSM. Although various approaches for indoor mapping have been developed, their usability is still debatable. Opposing to services for outdoor mapping, projects linked to buildings interiors are mostly based on volunteer activities of the nonprofessional community.

While we find numerous projects of indoor mapping related to OSM, there are also different activities by companies or other communities outside of OSM. Individual efforts try to respond to specific demands or they develop as global indoor mapping projects. Individual projects such as The Alpen Adria University Klagenfurt or Vienna University of Economics and Business Administration indoor routing provide navigation information

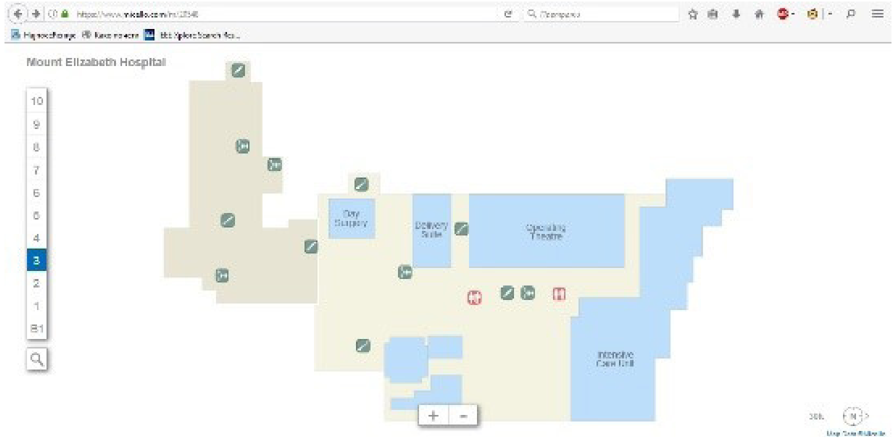
19 as per June 2017

on their campuses. Opposing to previous mentioned projects, services such as Anyplace, Google or Bing develop larger systems of global mapping. Anyplace or Google Maps users can also add/edit building floor plans and information.

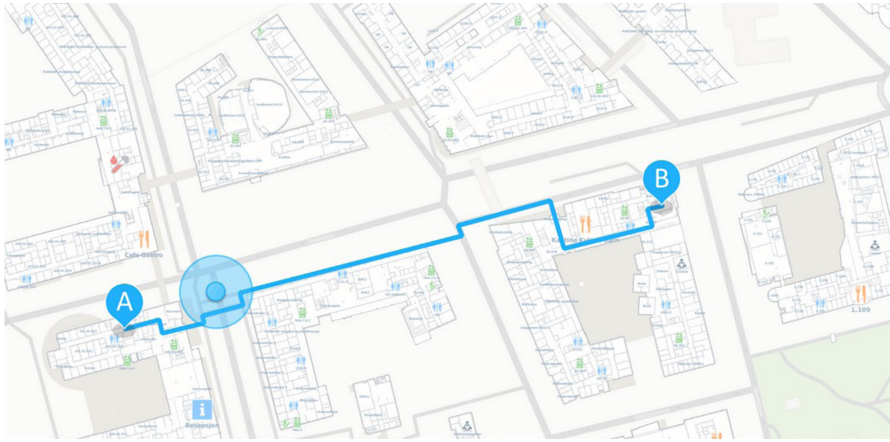
Summarising briefly, we found out that current projects for indoor mapping are focused on certain kinds of buildings such as airports, railway stations, museums, schools, shopping malls, parking decks and universities, but not hospitals. On the other hand, many hospitals around the world already have their own navigation systems, e.g. for way-finding in hospital (Figure 6).

For architects and hospital planners, such readily-available maps are crucial not only for simulation purposes, but also as basis for planning work.

This thesis thus tries to bridge the gap between what is available (non-digital plans) and what would be a major benefit for the community (digital plans/indoor maps).



Indoor Maps from Micello
Mount Elisabeth Hospital, Singapore



Indoor Maps from MazeMap
St. Olavs Hospital, Trondheim University Hospital

Figure 6 Indoor mapping in hospitals

2.4. Vectorization of Architectural Drawings

> Building plans use various types of graphical representations. There is no universal language of architectural plans. Even within a single office, different teams often have different graphic style. Besides that, there are several classes of drawings, in the range of sketch to final plans. Consequently, diverse drawing scales are used. Preliminary designs are usually drawn to a scale of 1:200 and final drawings to a scale 1:50 (recommended scale for component drawings is from 1:10 to 1:1).

The graphic style of drawing is closely related to scale. In the preliminary design stage, walls are commonly depicted with white, grey or black fill, having black borders of consistent thicknesses. As the project develops the walls' colour and line thickness changes depending on materialization. Most importantly, at some point, colours are also swapped with hatching, which significantly influence the appearance of the drawing. Although the use of particular hatches for different material types has been widely adopted, there is no a consensus on this topic.[20]

If it were possible to choose between various types of building plans for simulation, preliminary designs would be the best choice considering their visual simplicity: Designs at this stage contain only coloured walls, room descriptions, furniture and dimension lines to a certain extent. Plans at later stages are far more detailed: They might additionally contain layers for HVAC, electro installations and fire safety (e.g. sprinklers). The difference between preliminary plans and fully detailed ones is very important, since higher levels of complexity produces outcomes of less quality (details have to be deleted, parts of the plans re-drawn, etc.).

20 the argument that there are norms for graphical symbols and drawing styles is correct, however, the drawings we expect to scan might pre-date or partly neglect these norms (as is often the case in practice)

Over the past 30 years, various attempts have been made to increase efficiency of floor plan recognition systems. Floor plan analysis can be seen as a specialisation of image analysis: The aim is to identify and extract graphic elements.

The input is a raster image, produced by scanning or photographing architectural drawings. A quick survey of current approaches[21,22,23] shows that the scientific community is following either a comprehensive approach or undertakes restricted tasks referring to certain issues. These particular issues are text and graphic segmentation, as well as discovery of graphic symbols.

Recognition and extraction of hand-drawn or CAD geometric shapes begins with identification of basic elements such as lines, arcs and circles. This is followed by extraction of graphic symbols that consist of geometrical primitives. Recognition is usually executed by following a rule-matching model. Template-matching, attributed graph matching and network of constrains are some of the matching methods, that have been developed over the past decades. Regardless of the method used to match features, understanding of semantic relations embedded in architectural drawings as well as a certain domain-specific knowledge is very often inevitable.

Vectorization can also be performed by using commercial, user-friendly tools. Tracing i.e. recognition of objects is a built-in command by software such as Adobe Illustrator or Corel Draw. Imported data can be traced simply by choosing “image trace”. Additionally, it is possible to choose between several output options for graphic style (tracing result, tracing result with outlines, outlines, outlines with source image and source image) and pre-sets that provide pre-specified tracing options for specific types of artwork (technical drawing, line logo, high/ low fidelity photo etc.).

Software such as Vector Magic, Easy Trace, Potrace or

21 L. P. de las Heras, D. Fernández, E. Valveny, J. Lladós and G. Sánchez, “Unsupervised Wall Detector in Architectural Floor Plans,” 2013 12th International Conference on Document Analysis and Recognition, Washington, DC, 2013, pp. 1245-1249. doi: 10.1109/ICDAR.2013.252

22 Sébastien Macé, Hervé Locteau, Ernest Valveny, and Salvatore Tabbone. 2010. A system to detect rooms in architectural floor plan images. In Proceedings of the 9th IAPR International Workshop on Document Analysis Systems (DAS ‘10). ACM, New York, NY, USA, 167-174. DOI=http://dx.doi.org/10.1145/1815330.1815352

23 Chang, M.M., Or, S., Wong, K., & Yu, Y. (2005). Highly Automatic Approach to Architectural Floorplan Image Understanding & Model Generation.

ImageTracer are designed particularly for conversion of bitmapped images into vector graphics.

However, these are examples of general vectorisation packages, which have no semantic understanding of floor plans. In contrast to these, Scan2Cad is a specialised software that is able to recognize text layers, curves, arcs, lines, to detect corners, connect shapes and so forth; GTXImage CAD is a similar raster and vector drawing system for bringing legacy paper drawings into a modern CAD, Electronic Document Management (EDM) or GIS environment, disposing with 1500 vector commands.

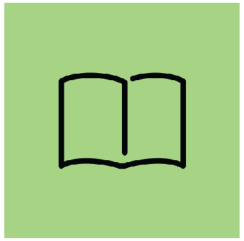
For recognition of architectural drawings from images, we will still use Potrace because it is free-source software and we do not need semantic recognition if plans are preliminary (thus already simplified - nothing to recognize apart from room boundaries).

1.

Introduction

2.

Computer Simulation
as a Planning Tool for
Architect



3.

Related Work

4.

Improved Workflow

5.

Case Study Baden

6.

Discussion

3. Related Work

3.1. Early-Stage Hospital Simulation

> Wurzer et al.[24] suggest simulation-based visualization in the early phase of hospital planning which facilitates a tender writing process and simultaneously helps planners, staff and clients to understand workflows. The simulation is built based on architectural schemata, which give approximate areas/locations for each space. They are developed at the preliminary concept stage where the form of the building is not fully formulated yet.

Wurzer's work generally uses patient flows given as spreadsheets. Such data are obtained from the Hospital Information System (HIS) or the underlying Enterprise Resource Planning (ERP) system. The capacity of spaces is given as an integer quantity stating how many patients can be served in parallel. The simulation computes the arrival and utilisation of these spaces over time, leading to the formation of queues (utilisation > capacity; queueing model). The results are visualized e.g. bubble diagrams[25] where the size of each bubble corresponds to the usage of the space and the size of each edge between two bubbles corresponds to the throughput. Additionally, properties such as the colour of a bubble may depicts qualitative measures (e.g. over-utilized, well utilized and under-utilized).[26]

Tabak[27] has developed an occupant simulation in which activities (e.g. meetings, presentations) of office users are simulated. The underlying basis and environment for the simulated occupants

24 Pre-Tender Hospital Simulation Using Naive Diagrams As Models, Wurzer G., Lorenz W.E., Pferzinger M., 2012, in Proceedings of the International Workshop on Innovative Simulation for Health Care, ISBN 978-88-97999-13-3, p. 157-162, https://publik.tuwien.ac.at/files/PubDat_209962.pdf, accessed 5 Mai 2017

25 Space Adjacency Analysis: Diagrammaing Information for Architectural Design. White E.T., 1986, Architectural Media.

26 From Quantities to Qualities in Early-Stage Hospital Simulation, Wurzer, G and Lorenz, W, 2013, in Proceedings of IWISH 2013, pp. 22-27.

27 User Simulation of Space Utilisation, Tabak, V, 2008, PhD Thesis, TU Eindhoven.

is given by a network graph, which is also used to compute the passage between two areas. The core of this approach lies in occupant schedules, which were surveyed and validated[28] for a real building. A simplified procedure for obtaining occupant schedules was later provided by Goldstein et al.[29], based on generating fictional schedules from calibration data.

In these approaches, the logic that simulated building occupants follow is a simplified (task-based) one. Simeone et al.[30] have provided a framework also for complex activities between multiple actors coming together for a task.

3.2. Recognition of Floor Plans

An important pre-step for simulation lies in the provision of proper plans, which form the environment for a model. These can either be re-drawn from scratch or obtained digitally. Apart from that, it is also possible to use drawing recognition for getting digital plans out of analogue representations. Many approaches have been developed over the years.

Koutamanis et al.[31] argues that recognition of digitized drawings can occur at the levels of geometric shapes, architectural elements and spatial articulation. The identification of geometric primitives such are lines or arcs is achieved by edge following and chain coding. Consequently, two-dimensional closed shapes that consist of these primitives are recognized. Architectural

28 Validating an office simulation model using RFID technology, Tabak, V, de Vries, B, Dijkstra, J, 2008, Proceedings of the 9th International Conference on Design & Decision Support Systems in Architecture and Urban Planning / Ed. H.J.P. Timmermans, B. de Vries. - Eindhoven : Eindhoven University of Technology, 2008. 15pages.

29 Schedule-Calibrated Occupant Behavior Simulation, Goldstein, R, Tessier, A and Khan, A, 2010, Proc. SimAUD 2010, 8 pages.

30 Simeone, D., Kalay Y.E., Schaumann, D., and Hong, S.W. (2012). An Event-Based Model to Simulate Human Behaviour in Built Environments, Proceedings of eCAADe 2012 (Volume 1), pp. 525-532.

31 Architectural computer vision: automated recognition of architectural drawings, Koutamanis A. And Mitossi V., 1992, in Proceedings of the 11th IAPR International Conference on Pattern Recognition, ISBN 0-8186-2910-X, p. 660-663, http://ieeexplore.ieee.org/document/201647/, accessed 8 Mai 2017

elements such as walls, door or windows show a wide range of visual representations. However, although there are no generally accepted standards, it is possible to identify some common visual characteristics, which may be applied on elements within one group. For instance, two long, parallel lines at a short distance represent most probably a wall. Stairs can be traced correspondingly, by detecting line of travel and its relation to stair treads.

Other architectural elements can also be classified according to predefined parameters. Additionally, it is possible to establish the most common graphical methods that are employed to illustrate architectural plans. Certain standardization of elements enables the use of recognition templates. A template is a prototype i.e. example of a known object. Recognition is therefore the task of matching image segments to pre-defined prototypes.

Lu et al.[32] have developed algorithms that identify architectural elements. This identification is based on internal semantic relations rather than visual features. Such an approach shows numerous advantages, especially if we take into account the number of possible elements and their graphic representations, which can appear in the drawing.

Their system initially detects components that are more regular and subsequently, based on domain knowledge, other architectural elements. For instance, it will recognize grid or dimension lines and afterwards columns as junctions of grid lines. It will discover beams (by following the columns) and so forth.

Floor plan sketches are generally characterized by positional ambiguity and shape distortion. Aoki et al.[33] have developed a prototype system that automatically converts a hand-sketched floor plan into a CAD drawing by using separate algorithms, one for line segment extraction and other for closed region extraction.

32 A new recognition model for electronic architectural drawings, Tong Lu, Chiew-Lan Tai, Feng Su, Shijie Cai, 2005, Computer-Aided Design Volume 37, Issue 10, p. 1053-1069, http://www.cs.ust.hk/~taicl/papers/CADrecognitionmodel.pdf, accessed 7 Mai 2017

33 A Prototype System for Interpreting Hand-Sketched Floor Plans, Yasuhiro Aoki, Aki Shio, Hiroyuki Arai and Kazumi Odaka, 1996, in Proceedings of the 13th International Conference on Pattern Recognition, ISBN 0-8186-7282-X, p. 747-751, http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=547268, accessed 7 Mai 2017

Shape distortion is corrected by template matching of closed regions, and positional ambiguity is resolved in nearest grid-location manner, considering that architectural plans are usually drawn along grid lines.

Text recognition i.e. room labelling is also an important part of the conversion from analogue to digital. Ahmed et al.[34] have developed a system that extracts both structural as well as semantic information from floor plans. Text layers are subjected Semantic analysis, which enables capturing of useful descriptions. Sheraz et al.[35] also propose a method for extraction of graphical elements from floor plans, by introducing a differentiation between thick, medium and thin lines/walls. The method helps to retrieve information of different types (bearing walls, self-supporting walls, symbols) in a simpler manner. It has been extended by feature that enables the elimination of components outside of the outer walls.

Template-matching based object recognition shows weaknesses, in terms that every template deviates to some extent from concrete drawings. If the template is not elastic, extraction will cause unpredictable results. Valveny et al.[36] thus propose a use of deformable template matching. An initial image/ template, that represents an ideal shape of a symbol, is altered conforming to predefined set of rules, aiming the adjustment to the target image, but preserving reasonable similarity to the ideal shape.

In addition to presented methods, there are also other prospects oriented toward acquiring building plans. Alzantot et al.[37] propose a crowdsourcing-based system (“CrowdInside”) for the automatic construction of buildings floor plans. The system uses the smartphones sensors to construct automatically accurate motion traces. CrowdInside receives and processes the motion traces from different visitors of a building to detect its floor-plan shape, number of rooms, their location, shape etc.

34 Automatic Room Detection and Room Labeling from Architectural Floor Plans, Sheraz Ahmed, Marcus Liwicki, Markus Weber, Andreas Dengel, 2012, 10th IAPR International Workshop on Document Analysis Systems, ISBN 978-0-7695-4661-2, p. 339-343, http://ieeexplore.ieee.org/document/6195390/, accessed 6 Mai 2017

35 Improved Automatic Analysis of Architectural Floor Plans, Sheraz Ahmed, Marcus Liwicki, Markus Weber, Andreas Dengel, 2011, International Conference on Document Analysis and Recognition, ISBN 978-07695-4520-2, p. 864-869, http://ieeexplore.ieee.org/document/6065434/, accessed 6 Mai 2017

36 Application of Deformable Template Matching to Symbol Recognition in Hand-written Architectural Drawings, Ernest Valveny, Enric Mart, 1999, in Proceedings of the 5th International Conference on Document Analysis and recognition, ISBN 0-7695-0318-7, p. 483-487, http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=791830, accessed 7 Mai 2017

37 Demonstrating CrowdInside: A System for the Automatic Construction of Indoor Floor-plans, Moustafa Alzantot, Moustafa Youssef, 2013, IEEE International Conference on Pervasive Computing and Communications Workshops, ISBN 978-1-4673-5077-8, p. 321-323, http://ieeexplore.ieee.org/document/6529506/, accessed 9 Mai 2017

1.

Introduction

2.

Computer Simulation
as a Planning Tool for
Architect

3.

Related Work

4.

Improved Workflow

5.

Case Study Baden

6.

Discussion

4. Improved Workflow

As already mentioned, the current workflow of asking for permission and scanning plans at the Magistrate requires an extensive amount of time that could be considerably reduced when resorting to photographing fire escape plans. We have tried to come up with a better approach in order to simplify the data collecting and preparation phase. Time efficiency in logistics processes ensures more time for data analysis, i.e. effort spent on the simulation itself.

Figure 7 compares current and improved workflow. As main contribution, we have reduced the number of necessary steps for obtaining digitised plans from seven to four. The rest of this section will argue why such a reduction is possible, and how such an approach helps saving time and effort.

The process itself is still not automated to full extent. If we had fully digitized drawing with separated layers, it would be very simple to develop a simulation from that basepoint. At this point, we still need to invest some certain time and effort in order to transpose input data to requested form.

Improvement in the process begins with the first phase – data collecting. Instead of applying for permissions by city authorities, we will photograph/scan building plans from available resources. Plan adjustment must be performed in both scenarios. If we hold on to the first scenario i.e. standard workflow, we have to crop, stitch and scale plans, especially if they are scanned with handheld scanner.

Second scenario i.e. improved workflow disposes with photographed/scanned plans. Cropping, stitching and scaling are again necessary. The next step shows a significant improvement. Standard workflow sees vectorization and add/complete features as the only logical proceeding. In distinction to standard workflow, improved one skips two previous mentioned operations. Instead, it offers a new one, where we perform a simple flood-fill operation on every recognized space. Vectorization performs later automatically after we upload the

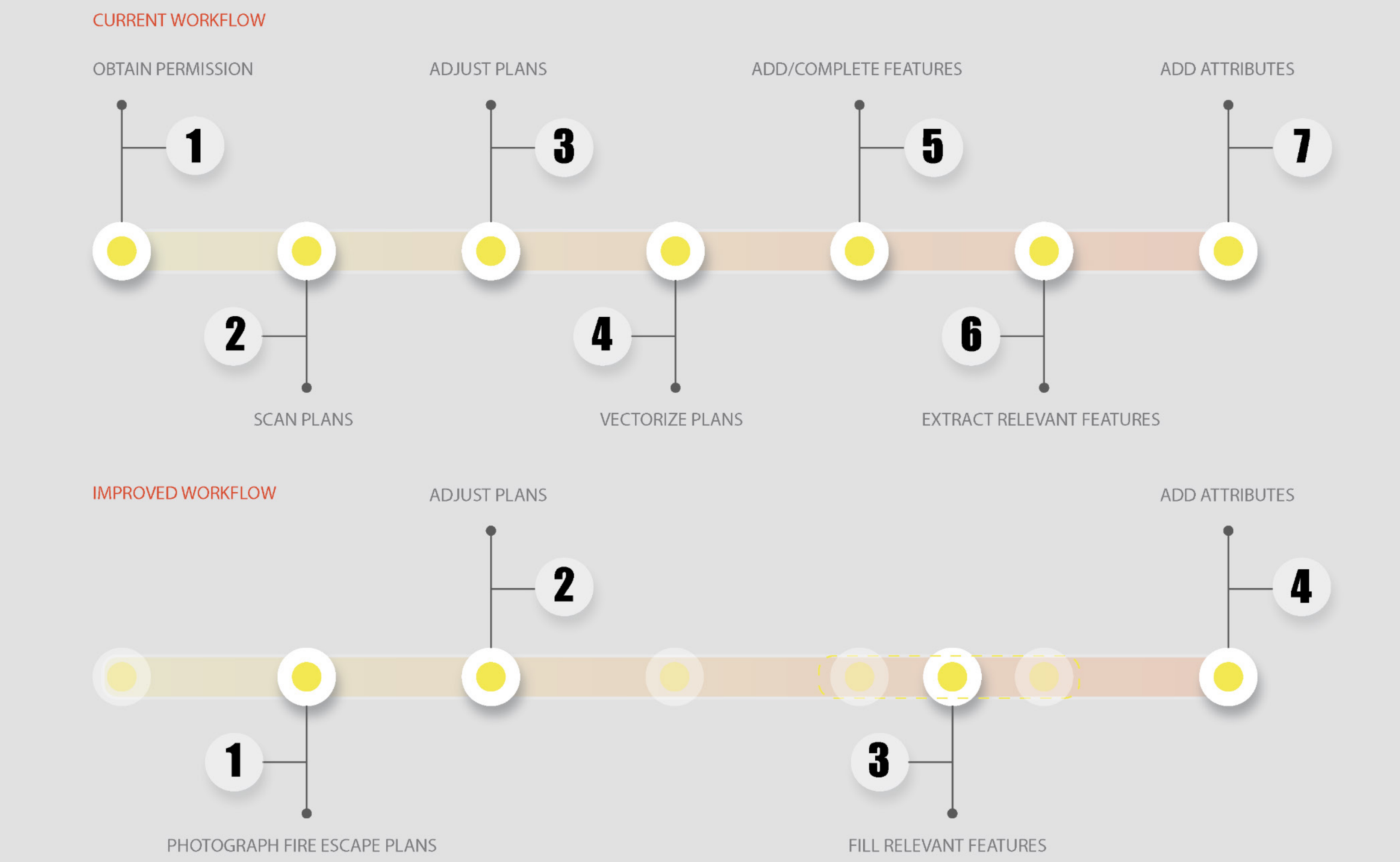


Figure 7 Current and improved workflow

plans into database. Additional features can be also added in the final step, either interactively or by uploading a spreadsheet. We can assign specific attributes to areas by making connection between colors and attribute for instance. The connection given in a spreadsheet will be than uploaded to the database.

4.1. Availability of Building Plans

> Simulation uses simplified building plans as an input. Multiple methods and tools can be used to collect plan data, depending on availability of resources, simulation purpose and type of information that we seek. From an architect's point of view, we can identify the purpose of simulation as a tool in the planning process.

If the phase in which the simulation acts as tool is set at an early point within the planning project, the information we need can be reduced to some basic architectural elements that roughly define a building, which comes down to schemata containing a collection of space polygons optionally carrying some attributes.

The standard workflow for obtaining such schemata for simulation is initiated with requiring a permission from appropriate authorities for viewing/coping building plans. This step usually requires more time than all other phases together. The process can be improved by photographing fire escape plans or by scanning/photographing drawings from literature. As early-stage simulation requires only attributed polygons (attributes e.g. in the form of colours stating the type of space), information provided by fire escape plans, books or architectural magazines is enough to reach such a representation.[38]

4.1.1. Scanning Building Plans from Books and

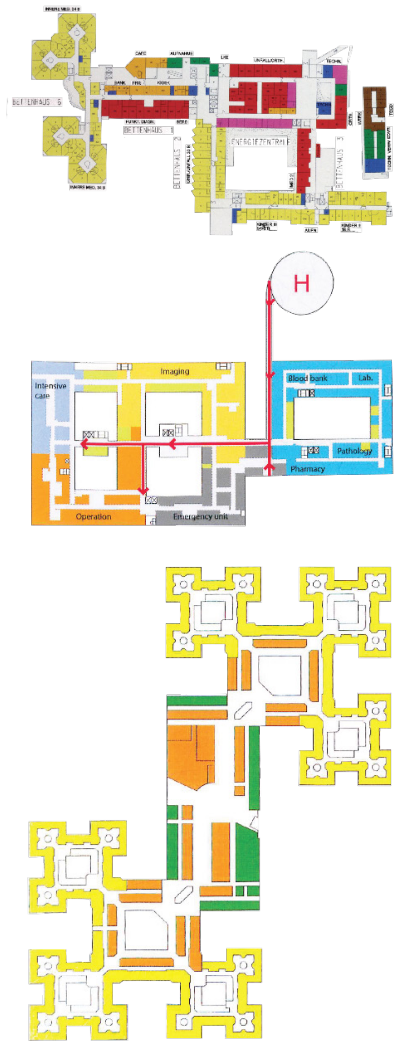
> It is not rare that architects present their projects through various printed publications. Some books elaborate only one project in detail, while the other ones are concerned with many designs (Figure 8).

The most complex issue concerning use of building plans from

38 As platforms for finding data, we use the internet and libraries. In addition, many books that contain architectural drawings are available in bookstores as well.

books and magazines is lack of data, as only the most interesting parts of a project and not the project as an ensemble are typically presented. This is especially true in the case of articles that present a number of projects of a common building type. In contrast, publications that include only one project usually offer a deeper level of detail. Furthermore, different drawing scales may complicate the process: Even if a book presents an appropriate building plan, the scale is often extremely small; enlargement of drawings without reference objects can lead to significant errors.

Apart from improper scale, different levels of detail in the presented content can lead to additional problems. For simulation, we do not need many details but only some characteristic information (i.e. outlines of space). When a publisher wants to publish a building plans at an extremely small scale (caused by a book's format), he/she needs to clean up the plans to keep readability.[39] Therefore, building plans from literature commonly give only filtered data which we can use only if we have some additional information that describe the plans. For example, if drawings do not contain room labels, we need some clarification



a) Satisfying level of details

Color-coding according to DIN 13080 makes an interpretation of plans much easier.

b) Low level of details

c) Extremely reduced drawing

Figure 8 Building plans scanned from books

39 otherwise, small parts as furniture or text will look like ink stains

Figure 9 Klagenfurt provincial hospital, Dietmar Feichtinger Architects

(either as a text or as a drawing). Without proper labels we cannot determine a function of rooms and hence we cannot use such plans for further (functional) analysis.

4.1.2. Online Resources

> Online resources often present filtered data instead of full building plans. Housing projects seem to be more accessible than other designs, nevertheless finding and downloading complete project plans is a real rarity. In contrast to housing projects, hospital plans are almost impossible to find online.[40] It is hard to determine if the cause for such information retention lies in the competitiveness of architecture firms or in not recognising the needs of a broader audience. All we can say is that hospital plans are not publicly available at this point, and that mapping services could thus benefit from a reverse-engineering approach as is presented in this thesis.

As already stated in chapter 2.3. there are efforts to map indoor spaces and make them globally online available. For now,

40 even if we include search engines of other countries and type in a hospital name in its local languages as well, outcome will be quite the same

most of these attempts are oriented toward some other types of buildings. Healthcare facilities however are increasingly a subject of custom-built activities on indoor routing and navigating systems. More and more hospitals in world develop navigating systems available as mobile apps but visible also from any other device connected to the internet.

As far as other online sources are concerned, we are able to find usually only segments with a few exceptions. Figure 9 illustrates a very detailed depiction of hospital floors plans. With known color-coding and all floor plans given, the hospital can easily be used for any kind of analysis or simulation. The problem is that we usually do not need any hospital but specific one, so these few examples are not especially helpful for an aimed simulation. The further question that relates to printed literature as well is up-to-datedness of drawings. We can eventually find a plan either online or in book, but simultaneously arises the question is the building changed in the meanwhile and to what extent? If so, we cannot use the plans or we have to establish in any accessible way if the project was changed.

4.1.3. Photographing Fire Escape Plans

> Some public buildings are mandatory equipped with fire escape plans. These are usually gathering places such as theaters, building of special purpose such as hospitals or schools and any other high-risk buildings with or without public access. Evacuation plans are placed in rooms (hotel rooms, classrooms, hospital rooms), in hallways and staircases.

Exact position is commonly noticeable spot: main access to building levels, nearby staircases and elevators, hallway junctions and other meeting points. Additionally, pavilion-type hospitals usually have an orientation plan placed near the main entrance to complex (Figure 10) and block type structures poses such a plan nearby main entry to the building (Figure 11). The idea to extract visual information from fire escape plans comes from the fact, that those plans are accessible in contrast to building’s whiteprints. They are easy to find, always placed on

Figure 10 Orientation plan, Wilhelminenspital

prominent spots and reachable for everyone. The problem are only restricted areas. Hospital departments such as psychiatry are sometimes locked (although after recent studies on risks of suicide attempts in hospitals with locked-and open door policies, development takes the second course).[41]

We can use orientation plans as a base point. Hospitals built as blocks are easier to manage in terms of orientation. There is one main entrance in the building and the orientation plan nearby. Figure 11 depicts the orientation plan in Vienna General Hospital (AKH). Figure 12 shows single floor plan overview. The lack of accurate plan scale as well as extremely reduced level of detail obstruct the direct use of these plans for simulation purpose.

However, having in mind that the main orientation plan is quite simplified (it main task is to depict main departments in relation to our current position), floor plan overview discovers accurate location of elevators and staircases. Knowing that fire escape plans are situated close to the stairs, it help us to find those very quickly. Otherwise, we could spend hours only by looking for evacuation plans. Different countries have their own legal

40 Suicide risk and absconding in psychiatric hospitals with and without open door policies: a 15 year, observational study; Dr Christian G Huber, Andres R Schneeberger, Eva Kowalinski, Daniela Fröhlich, Stefanie von Felten, Marc Walter, Martin Zinkler, Prof Karl Beine, Prof Andreas Heinz, Prof Stefan Borgwardt, Prof Undine E Lang, 2016, The Lancet Psychiatry Volume 3 No. 9, p.842-849, http://www.thelancet.com/journals/lanpsy/article/PIIS2215-0366(16)30168-7/supplemental, accessed 25 June 2017

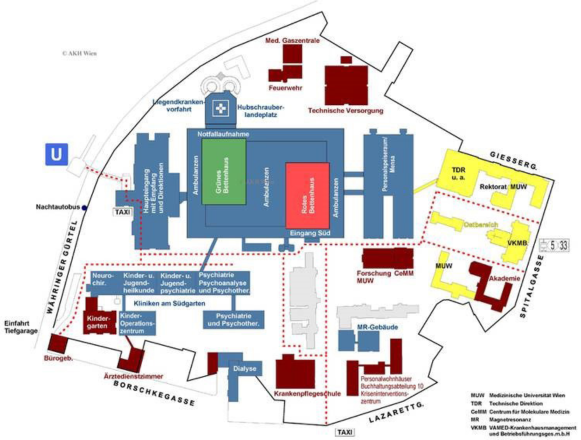


Figure 11 Orientation plan, AKH Vienna

Figure 12 Floor plan overview, AKH Vienna

regulations that control visual appearance and content of the plans. Visual representation of escape and rescue plans in Austria is based on DIN ISO 23601 i.e. after the new naming ISO 7010 because there is no Austrian standards or guidelines relating to the subject (on the other hand, production of fire safety plans is defined with TRVB 121).

Plans include standpoint, main and second escape route, location of fire extinguisher, fire hose reels, fire alarm, title, legend etc. Minimum size of escape plan is 297 mm x 420 mm (A3), except those in rooms by which the size can be reduced to A4. A scale of plans for big buildings is 1:250 and 1:100 for small facilities.

Background color is white or phosphorescent white. Every fire escape plane should contain also a general layout of a building, except those detail plans that depict already whole target facility.

Very important feature of fire escape plans is their simplicity. As shown on Figure 13, all unnecessary details are removed. Space utilization is the only information given i.e. written inside of a single room. In the next phase, when we need to adjust plans, it will be easy to a) remove the text or b) simply ignore it,

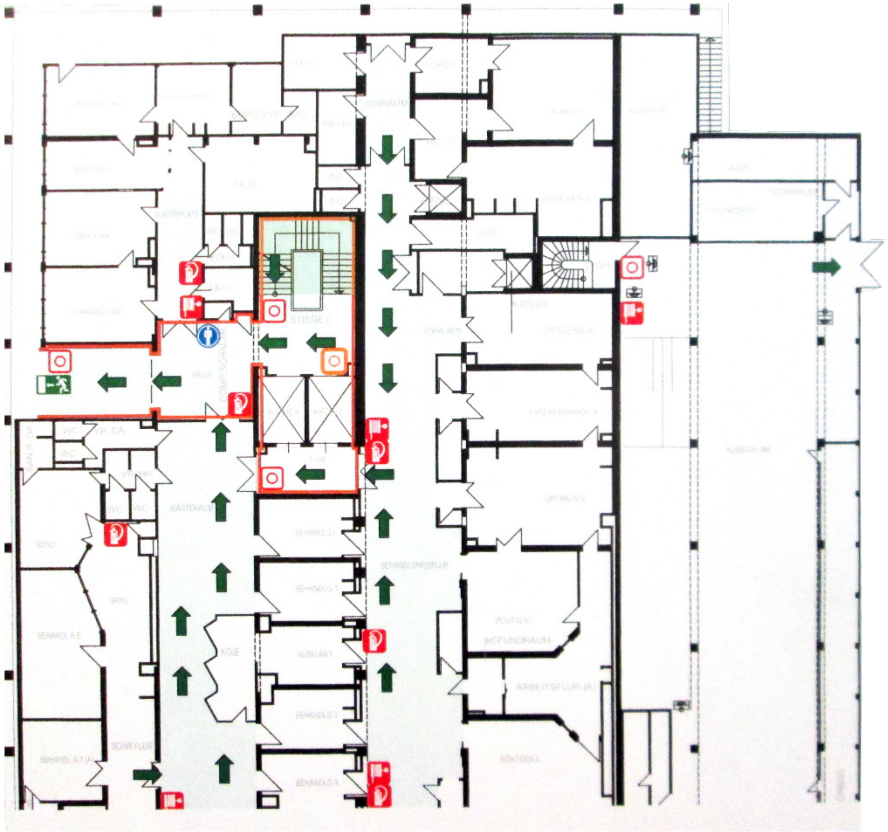


Figure 13 Fire escape plan, Wilhelminenspital, Trauma Surgery Pavilion

depending on a software that will be used.

If we use the paint bucket tool, in Adobe Photoshop for example, we can firstly remove all unnecessary features inside of one room. If we decide to draw polygons in a vector graphics editor, such as Adobe Illustrator, we do not need to remove anything from the plans. Polygons can be simply drawn on the plans and after the task is completed, plans can be hidden or removed.

4.2. Adjusting the Plans

After collecting i.e. photographing, all building floor plans need adjustments. Fire escape plans are vertical (hanged on the wall) and accordingly it is easy to keep the camera relatively parallel to the image plan. If we use plans from books the task is even easier, because scanner will ensure solid digitization. However, some deviations are inevitable and we must correct those.

Figure 15 depicts typical deviation caused by improper camera angle. Even the scanner will produce the same deviation because it is usually impossible to unfold a book in a proper manner, depending on binding type. We can use any raster drawing application in order to correct these distortions, by simple skewing and stretching. Removing unnecessary data is the next step. Fire escape plans are, as already described, extremely simplified in relation to standard project or submission plans.

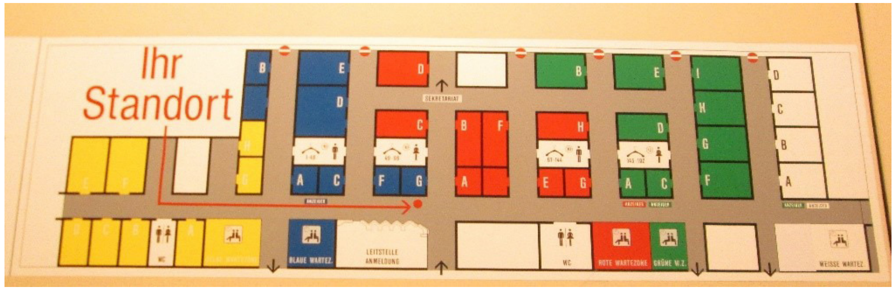


Figure 14 Orientation plan

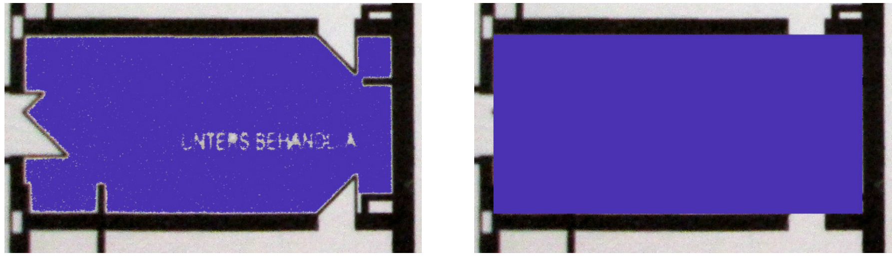


Figure 15 Applying a flood-fill operation on a) unaltered plan b) plan cleaned from unnecessary features.

After collecting i.e. photographing, all building floor plans need adjustments. Fire escape plans are vertical (hanged on the wall) and accordingly it is easy to keep the camera relatively parallel to the image plan. If we use plans from books the task is even easier, because scanner will ensure solid digitization. However, some deviations are inevitable and we must correct those.

Figure 14 depicts typical deviation caused by improper camera angle. Even the scanner will produce the same deviation because it is usually impossible to unfold a book in a proper manner, depending on binding type. We can use any raster drawing application in order to correct these distortions, by simple skewing and stretching. Removing unnecessary data is the next step. Fire escape plans are, as already described, extremely simplified in relation to standard project or submission plans.

Nevertheless, some details such as room ID need to be removed in case we use raster image editors (Photoshop for instance) for further editing. Simple fill operation, performed in Photoshop with paint bucket tool is shown on the Figure 15 If we use original, unaltered plans, result is quite fuzzy and bumpy, as shown on Figure 15a. Removing unnecessary features, in this case door openings and room label, we achieve a much better result (Figure 15b). Erasing irrelevant information is a simple task, performed just by choosing a specified area with rectangular selection tool, deleting its content with operation delete and applying the paint bucket tool afterwards.

Not only Photoshop, but also many other software packages such as Corel PaintShop, InkScape, Gimp or Paint have built-in tool displayed as a paint bucket. Behind this small tool stands a flood-fill algorithm that is used to connect neighboring and related elements of an array.[42] There are several types of algorithms, but all of them use a queue or stack data structure. Every algorithm takes following parameters: a target color, a replacement color and a start node. They can be also implemented in numerous programming languages, such as C, C++, C# etc. On the web, we can find many examples of algorithms implemented within different platforms. However, laity without even basic programming skills cannot make any use of it. It would be necessary to create graphical user interface (GUI) in order to facilitate use of the algorithm, which is not an easy task. Without GUI, a user must know exact color and coordinates of targeted pixel. The advantage of Photoshop and similar software packages is that everyone can use simple tools such as paint bucket even without serious previous knowledge.

42 Flood-fill algorithms used for passive acoustic detection and tracking, Eva-Marie Nosal, 2008, New Trends for Environmental Monitoring Using Passive Systems, ISBN 978-1-4244-2815-1, http://ieeexplore.ieee.org/stamp/stamp.

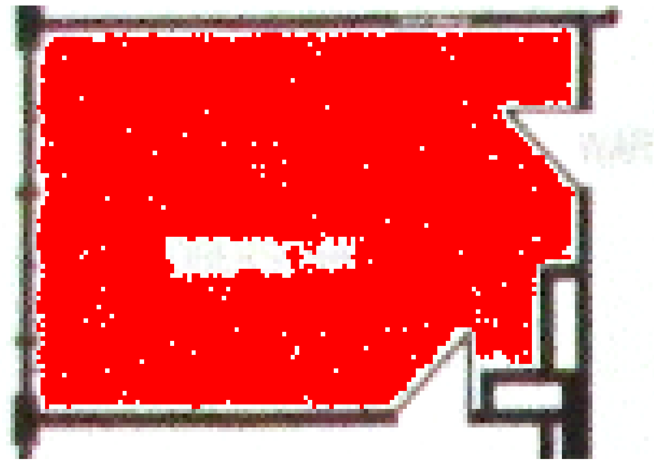


Figure 16 Execution of a flood-fill algorithm in C#

Figure 16 shows one room colored by using C#. However, as a result we need only room polygons, so if we decide to use C# for instance, then we need also to extract colored polygons from the plan. From all these reasons, it is much easier, maybe even for programmers, to use one-click solutions such as Photoshop. After polygons are obtained with flood-fill operation, follows the vectorization, which will be performed by using the open-source tool, Potrace.

4.3. Vectorization

In the previous step, one had to use flood filling in order to get all room polygons. At this stage, it was necessary to have a legend i.e. to assign a specific color to every single hospital department. As we said before, there is already a standard in form of DIN 13080 (Figure 17), where for each area of activity stands one color.

The legend can be extended, depending on what we want to show. For instance, instead of one red tone that depicts all departments related to examination, we can use several red tones in order to label separately every single department.

All departments should be listed within one spreadsheet and one color, specified as an RGB or HEX value assigned to each department. The vectorization, which follows in the next phase, uses Potrace algorithm. Since Potrace handle only 2-valued images, we need to a) convert colored schema into black and white image and b) keep in mind that all polygons should be

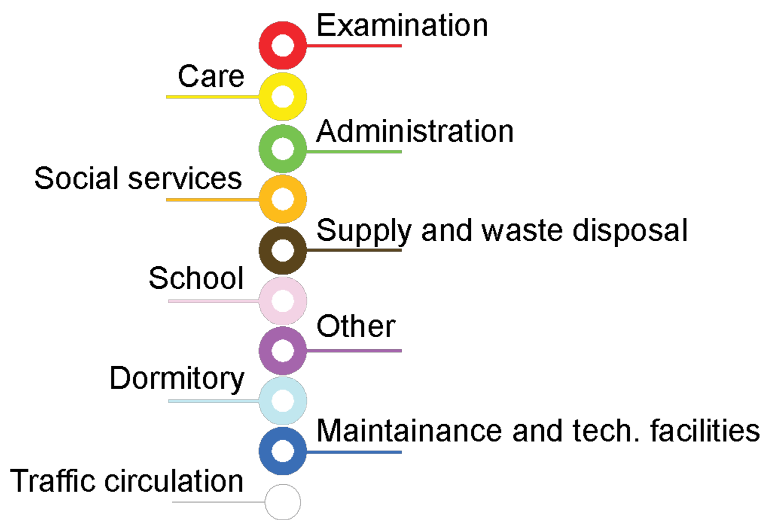


Figure 17 Functional areas according to DIN 13080

separated by at least one pixel, so that the conversion maintains room boundaries. However, there are also possibilities to use Potrace in processing color images. IncScape for instance has a built-in Potrace engine that works also with color images.

Figure 18 illustrates a vectorization process. Since we are working with database, colored schema will be uploaded to the database and vectorization follows automatically. In the process,

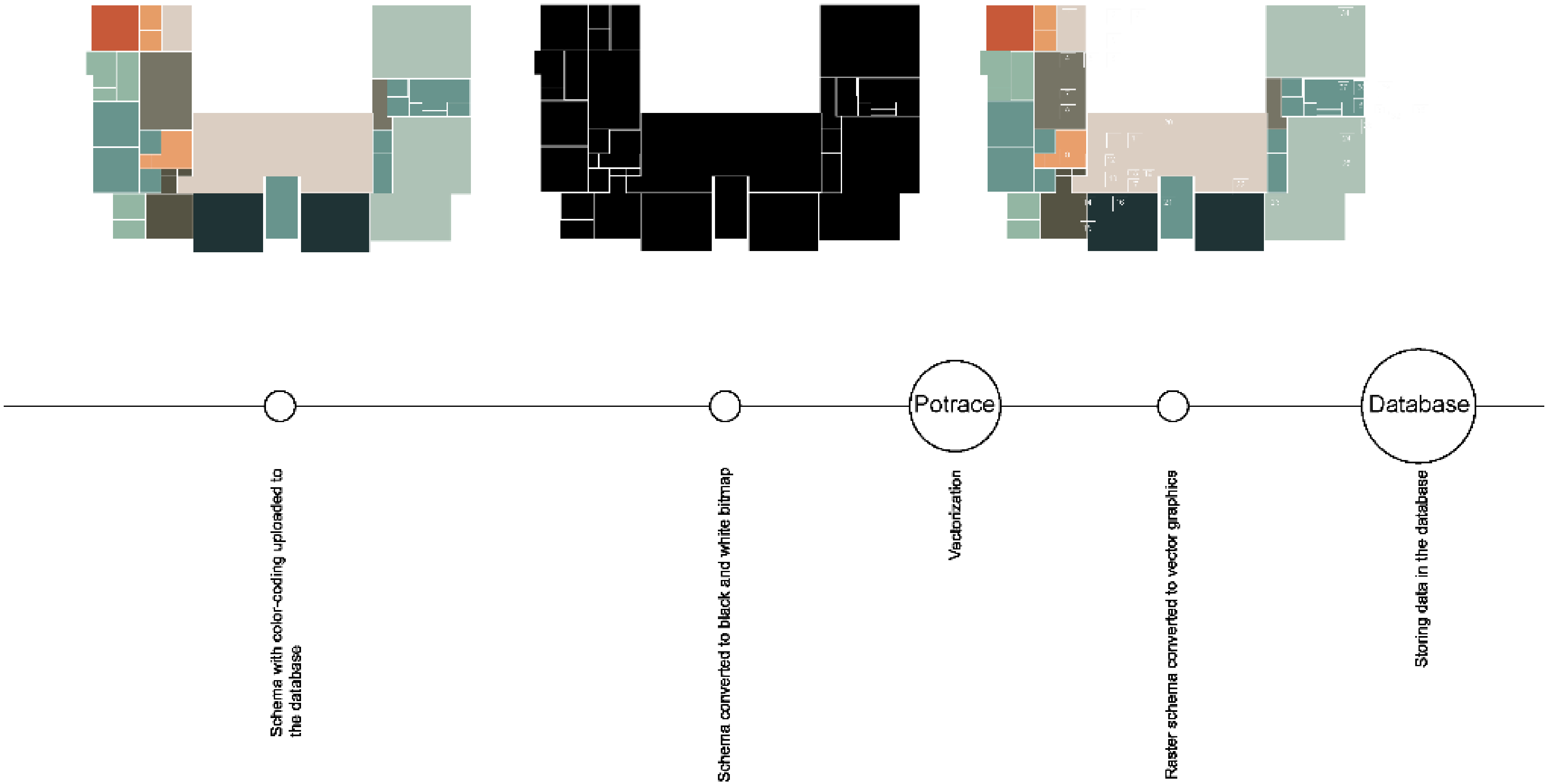


Figure 18 Vectorization process

every converted room polygon receives correspondingly ID and color. The spreadsheet that contains department's names and their colors will be also uploaded to the database. By writing code for forthcoming simulation in NetLogo, all necessary data we call directly from the database.

The other possibility would be to import manually every table and scheme in NetLogo, but database is quite useful and efficient considering that it allows easy access and analysis of data for each party by using web services for example. Change that has been made in database is valid for all users so it saves time and effort especially when it comes to teamwork.

By using spreadsheet, we can use colors to identify single departments. Considering that one may request single room as a functional area and not department as a whole, we can make use of the polygon IDs, assigned during the vectorization process. In that case, a spreadsheet containing room IDs and some attribute such as room label should be uploaded to the database.

Room labeling, as quite handy feature could be also obtained by using optical character recognition (OCR). It would be a faster process than making a spreadsheet with all labels, which is extremely exhausting task (a single level of a hospital may contain a few hundred of rooms).

4.3.1. Potrace

> Vector graphics or geometric design is a way of displaying images using geometric shapes such as points, lines, curves and polygons, which are based on mathematical equations. In vector graphics image is represented and stored as a collection of figures, together with their data (parameters) that define how the figure to be drawn and where it will be located. It is known by the name of Loss Less Graphics (graphics without losses). As the vector graphic is not based on a grid we can it infinitely increase or decrease without losing the quality. This is because it's based on mathematical functions, i.e. the absolute distance between the points.

Raster graphics or bitmap is a data defined as a regular rectangular mesh of cells called pixels, each pixel containing a color value. Only two parameters, the number of pixels and the information content (color depth) per pixel characterize them. The total number of pixels (resolution) and the number of values for each pixel (color depth) determine the quality of a raster image. If the color depth is greater, more shades can be displayed, which means that it gives a better picture and credible representation of the same. Pictures require a lot of memory, and therefore use different types of compression. Each color of each pixel is specifically defined so that (for example) RGB - Red Green Blue image contain three bytes for each pixel, and each byte contains a specially-defined color.

Raster image cannot be enlarged without loss of quality, which is not the case with vector graphics. Raster graphics are more practical than vector graphics for photographers and ordinary users. Vector graphics used by graphic designers and DTP (Desktop Publishing) editors. Rendering is the process of transposing vector images in raster form and it is rather simple. In contrast, vectorization (rendering of the raster to vector image format) is a pretty big deal, because it demands following the contour and shape of the bitmap, based on the difference color pixels.

Most known editor for raster images is famous Adobe Photoshop. Of course, there are many other: PhotoStudio, GIMP, GrafX2, ImageMagick, Ulead PhotoImpact etc.

Potrace (Figure 19) is a program that converts raster images into vector graphics. This process takes place in three steps and a possible fourth. In the first step, bitmap is dissolved in paths that divide black from white areas. In the second step, each path is approximated by a corresponding polygon. In the third step, each polygon is converted into a Bezier curve. In the fourth step, the resulting curve is being optimized by joining consecutive Bezier curve segments together where this is possible.

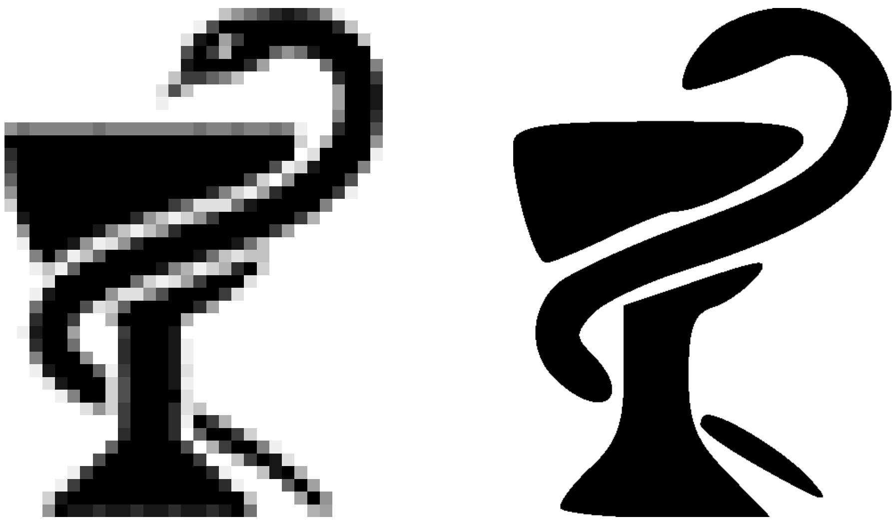


Figure 19 Original image (left), Potrace output (right)

Starting Potrace on Windows will occur from command window. (Figure 20) The easiest way is to simple navigate to folder where are potrace.exe and mkbitmap.exe., open the command window from this directory and type as shown on the Figure 20.Creating the path is the first assignment of Potrace algorithm (Figure

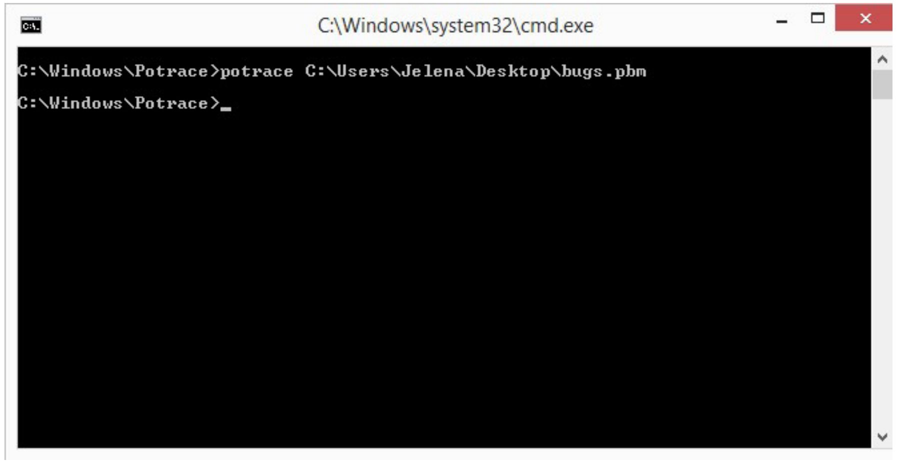


Figure 20 Potrace command window

21). The process starts from the premise that the bitmap is set in a coordinate system, where the corners of each pixel have integer coordinates. Point p is adjacent to four pixels, has integer coordinates and it´s called vertex if fours pixels don´t have the same color. Now let´s take two vertices v and w. Between v and w exist an edge, if the distance between v and w is 1, if an edge splits black from white pixel, in such way, that black pixel is on the left, and white on the right side when traveling the distance from v to w. Further, on, the path will be continuous sequence of edges. We say that the path is closed if the first vertex v0 is equal to the last one vn. Looking for the path will begin by

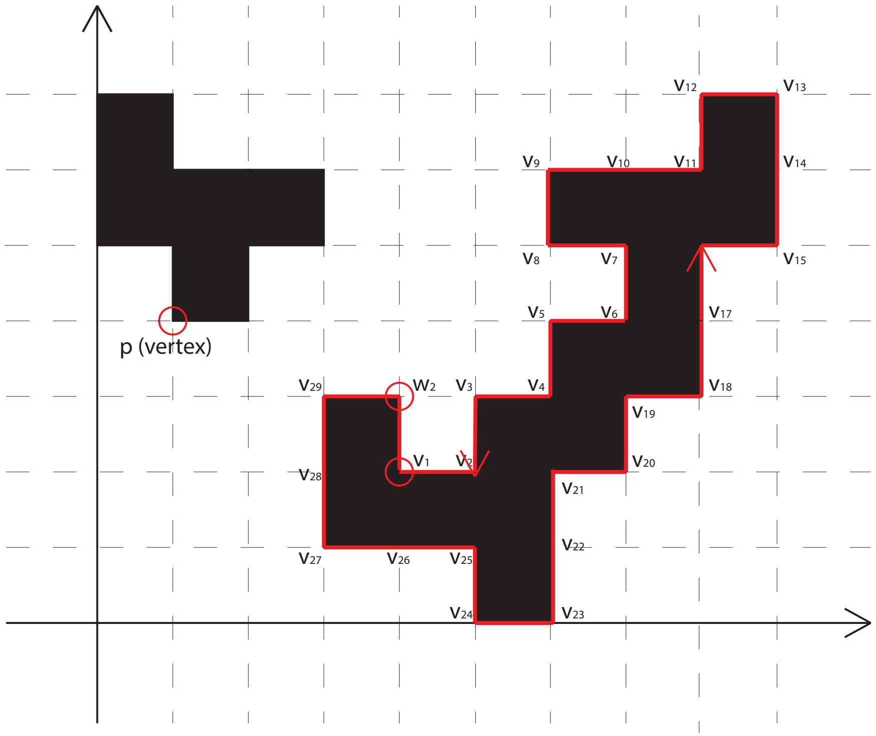


Figure 21 Original image (left), Potrace output (right)

searching a two pixels with different colors. Between them exist an edge. Potrace will determine the direction of the path by observing the edge so that the black pixel is placed on the left side and the white pixel on the right side of the edge. Afterwards the program just follows the path in such a way that the black and white pixels stay always on specified sides of the edge. The result is a set of closed paths to be passed to the next phase of the Potrace algorithm.

The Figure 22 shows path extension method. Traveling from one vertices to another along the edge, in such a way that black pixel stays on the left, and white pixel on the right side, we meet a choice on each corner: it is possible to take a right or left turn or, as shown in second option, the path expand by proceeding straight. However, there is also a fourth option-when the path could be extended on the right but also to the left. Path will be successfully created, no matter which turn will be taken. What would be different is for sure the shape of the path.

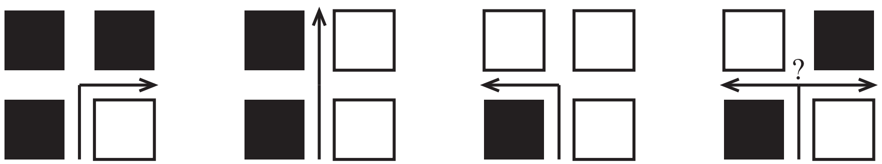


Figure 22 The path extension

Let us observe two points *a* und *b*, which do not necessarily have integer coordinates. Line segment *ab* approximates the path *p* $(v_0,...v_n)$ if $d\ (v_0,b)\leq 1/2$ and $d(v_n,a)\leq 1/2$ and for each $i=1,...,n-1$, there exists some point c_i on *ab* such that $d\ (v_i,c_i)\leq 1/2$. (Figure 23).

For a path $p=\{v_0,...,v_n\}$, we say the direction at index *i* is $v_{i+1}-v_i$, where $i=0,...,n-1$. There are four possible directions:

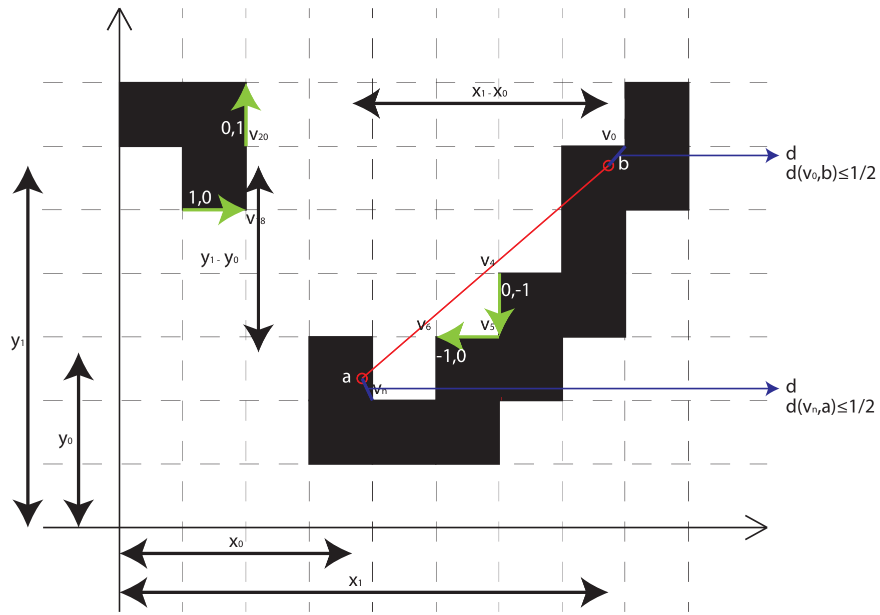


Figure 23 The path extension

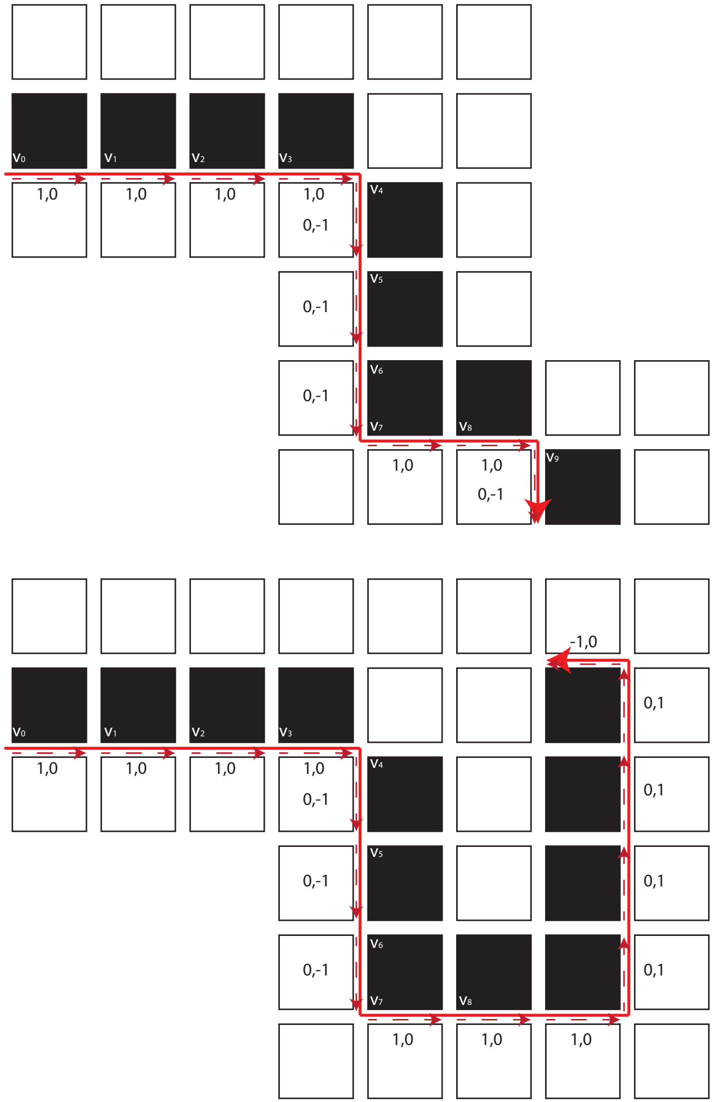


Figure 24 Straight and non-straight path: a) straight path b) non-straight path

(0,1),(1,0),(0,−1), and (−1,0). A path is called straight if some line segment approximates it and not all four directions occur in *p* (Figure 24).

Every closed path should be divided into segments. The path length from v_0 to v_n we can split into several sub-paths from v_i to v_j, where i,j є {0,...,n−1}. The subpath p_{ij} is $v_i,...,v_j$ if i≤j, or $v_i,...,v_{n-1},v_0,...,v_j$ if j<i. We can write jϴi for the cyclic difference between i and j, which is defined as jϴi=j−i if i≤j, and jϴi=j−i+n if j<i.

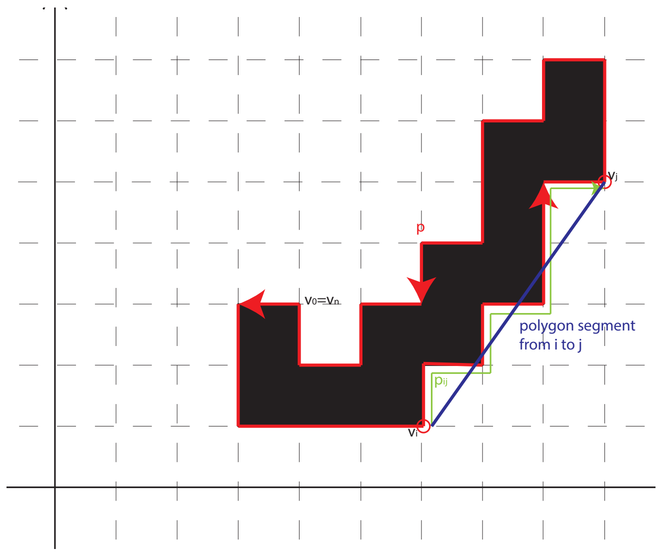


Figure 25 Creating a segment

In order to construct a polygon from a closed path *p*, we will define a possible segment from i to j if jϴi≤n−3 and the subpath $p_{i-1,j+1}$ is straight. A subpath corresponds to a possible segment if it can be extended by one point in both directions and still be straight (Figure 25).

It is clear that the algorithm is able to create a different polygons that corresponds to the same subpath (Figure 26). In that situation, it is necessary to choose an optimal one. The optimal

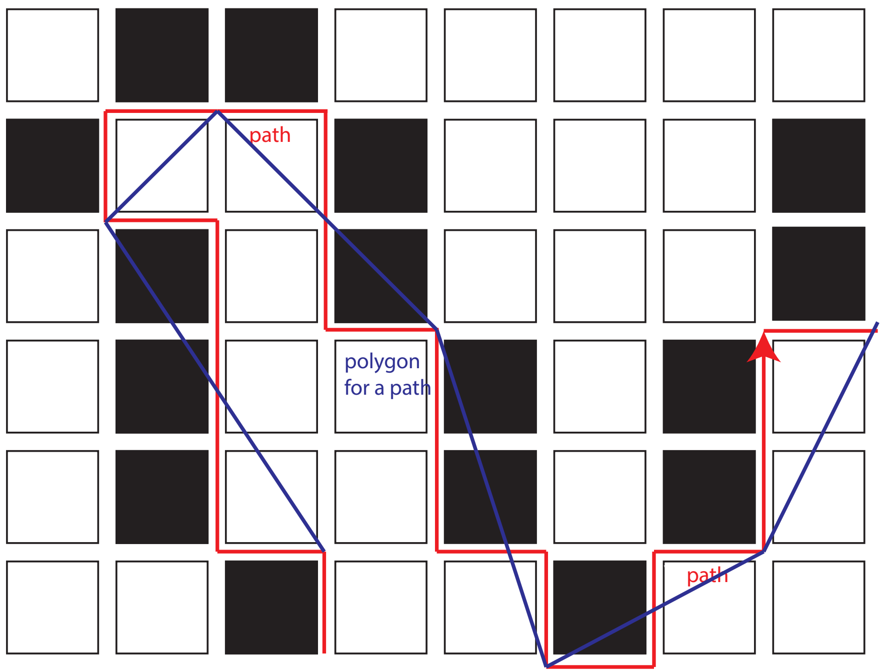


Figure 26 Creating a polygon for a path

one would be usually the one with the fewer segments but not necessary. Sometimes such polygons are not precise enough and show a great deviation from a path. In such a circumstances, we associate every segment with so called penalty. It is equal to the Euclidean length of v_iv_j, times the standard deviation of the Euclidean distances of the path points from v_iv_j.

Standard deviation is a measure of how much variability there is among a collection of individuals on one trait. It indicates what a typical amount an individual deviates from the mean of that collection, or you might say the typical distance an individual is from the mean.

On the other hand, we are dealing here with the Euclidean space and the deviation of a single point (vertex) from a line segment v_iv_j (Figure 26). In order to describe a penalty and his value we need to calculate a standard deviation of Euclidean distances between points and line segment. The value is going to be greater as the path point strays from the segment.

Now we have the path and the corresponding polygon. The polygon consist also from segments in such a way that every point of a path v_k has a corresponding vertex i_k on the polygon.

In the next step we will associate to each vertex i_k a point a_k in the coordinate plane not necessary of integer coordinates, such that a_k is near the v_{ik} and such that, for any two consecutive vertices i_k and i_{k+1} of the polygon the resulting line segment $a_k a_{k+1}$ is reasonably close to subpath $v_{ik},\ldots v_{ik+1}$. As a result, we have a polygon with adjusted vertices that we can call $a_0 a_k$.

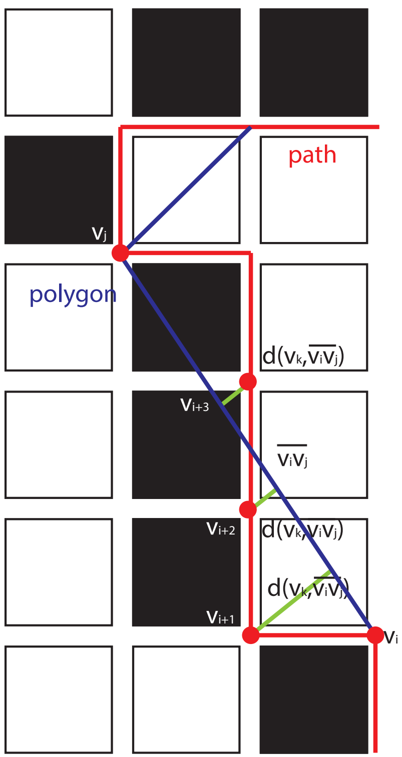


Figure 27 Deviation of a single point from the line segment

The last step is to convert a polygon from the previous step to a smooth vector outline. At this point we need to define Bezier curve (Figure 28). The simplest definition would describe it as curve that runs from some start point to the end, with its curvature influenced by one or more "intermediate" control points. Intermediate points are result of linear interpolation.

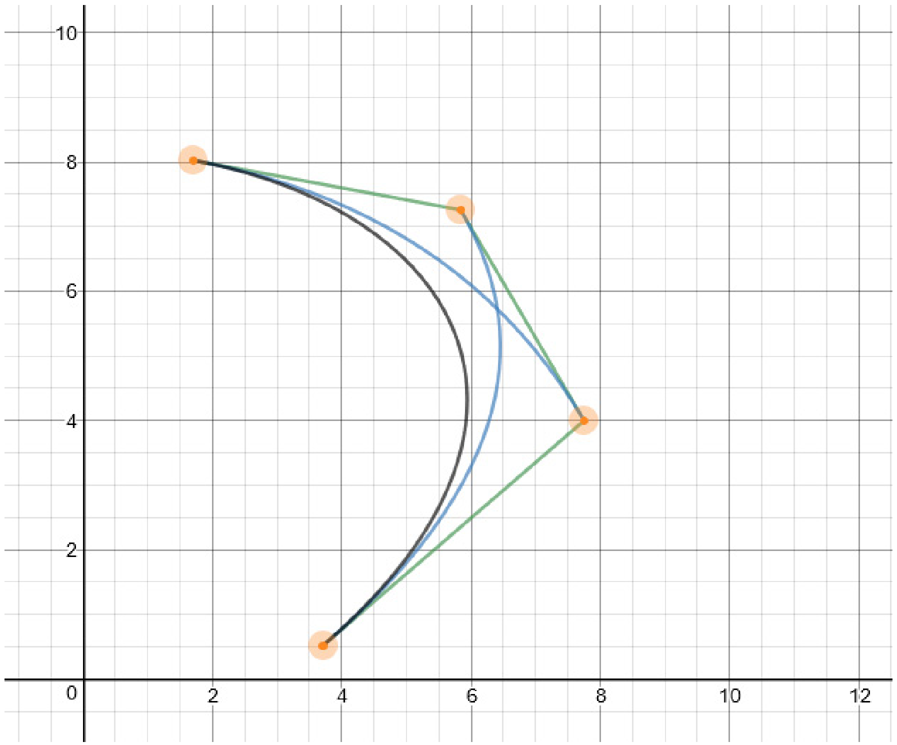


Figure 28 Bezier curve

That is nothing else, but inserting a new point between two points that are already given. We will observe one curve with four control points, z_0 z_1, z_2 and z_3. Also, we will restrict ourselves to convex curves, which means that z_1 lies between z_0 and 0, and z_2 lies between z_3 and 0. If we place a particular curve in coordinate system, the point z_0 will have coordinates (-1,0), z_3(1,0) and 0(0,0). Now let interpolate the points z_1 and z_2. Any Bezier curve is uniquely defined with following two parameters α,β є [0,1] such that z_1 (-1+α,α) and z_2 (1-β,β) (Figure 29).

Figure 30 depicts Bezier curves for all values of α and β that are multiplies of 0.2. As it seems, Bezier curves in every horizontal row are visually extremely similar expecting cases when α and β are very close to zero. Because of that we can restrict the problem by saying that α = β, which simplifies the task of finding an optimal curve.

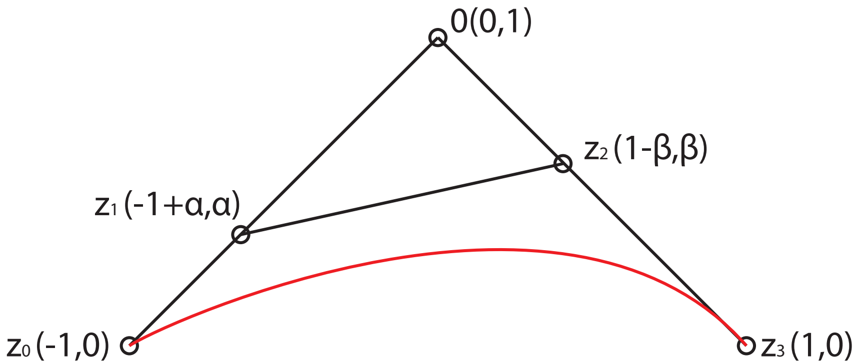


Figure 29 Process of interpolation

Figure 31 is showing the last step in the process of converting a raster to vector graphics. In the previous step we found a polygon a_o,..a_k. Let b_o,...b_k be the midpoints of the edges of the

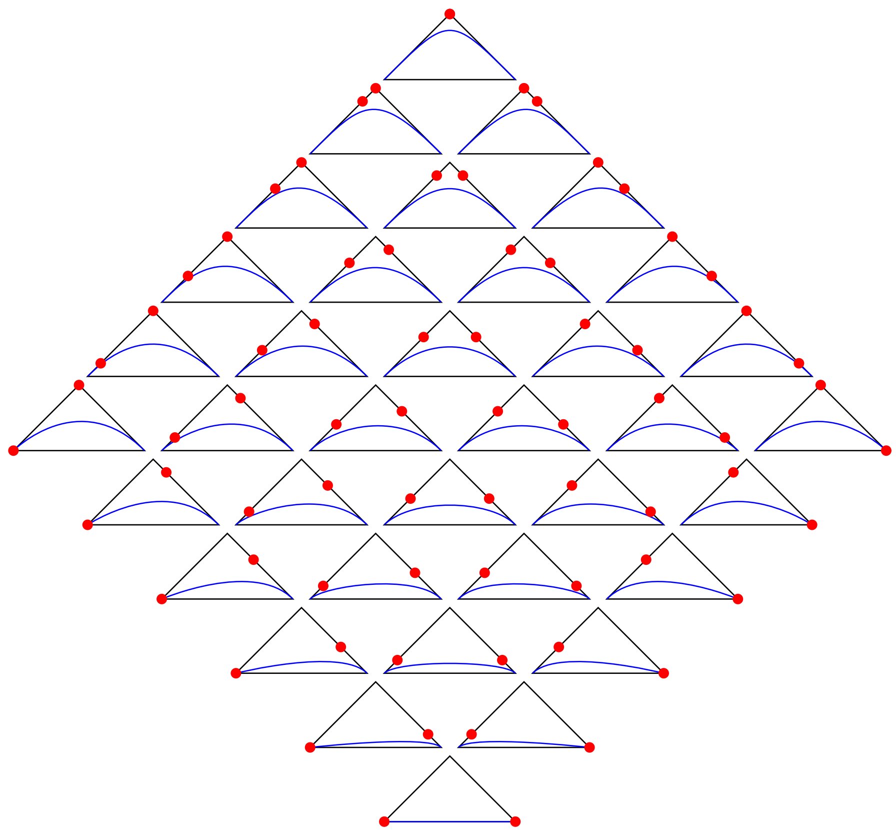


Figure 30 A 2-parameter family of Bezier curves

polygon. Next, we will construct the unit square around a_i and the line L_i, that is parallel to $b_{i-1}b_i$ and touches the square. The resulting Bezier curve will be so constructed, that it has tree tangents: a_ia_{i-1}, a_ib_i and L_i.

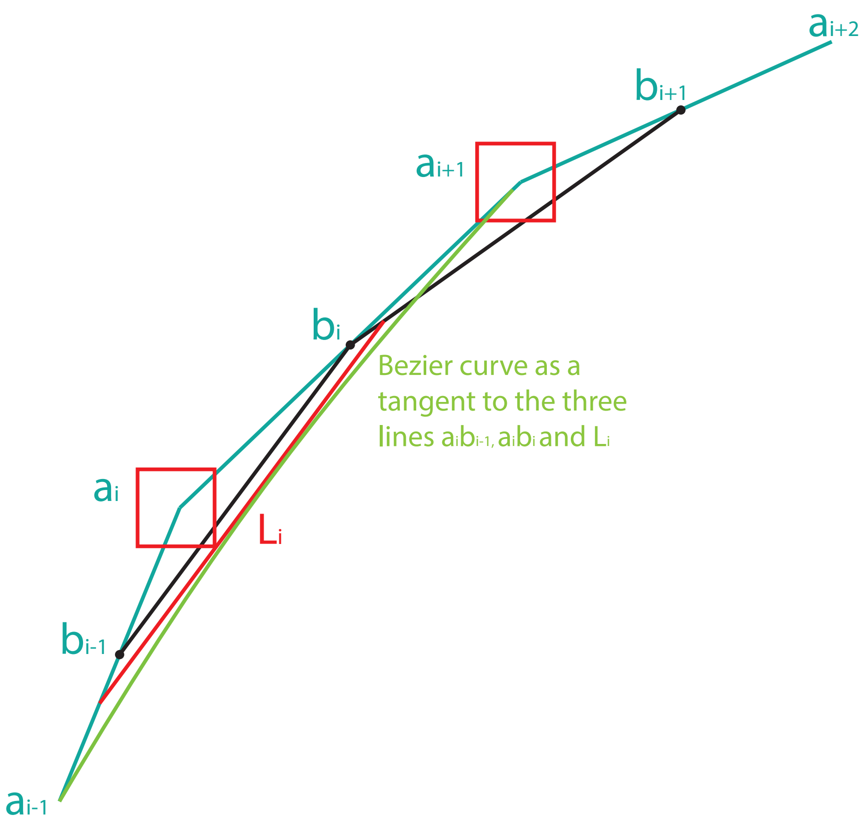


Figure 31 Smoothing

4.4. Presets

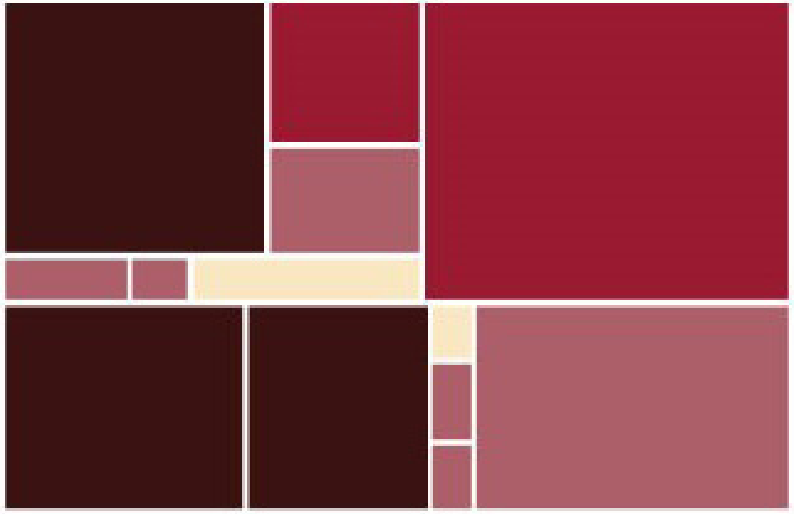
> Many simulation software packages have drawing tools or model libraries at least.[43] The same way, many drawing packages provide the possibility of simulating/animating in/of a built structure.[44]

If our main goal is simulation and not building model per se, we will use software package such as NetLogo, developed primarily for simulation. Although NetLogo gives the possibility to build the environment as well, it is much easier and faster to import a layout. That way we can be sure that spatial relations remain consistent. From that reason, we will import in NetLogo functional schema that has been previously vectorized.

Besides schemata, another pre-defined data can be used. As shown on Figure 32, schema consists from colored polygons separated by at least one pixel. When an agent travels from one department to another, he uses colors of polygons. Although we program his route of travel in such manner that an agent crosses a minimal distance, it is also important to define exact points where he can pass from one room to another. Only that way his movement is credible i.e. he is not walking through the walls. There are different ways to define points where agents cross from one room to another but probably the easiest way to do that is to import pre-defined layout of door openings in NetLogo.

Information such as people flow between departments or room IDs can easily be written in a form of a spreadsheet. NetLogo can read this spreadsheet if saved in a proper format such as .csv for example, which is much faster process in comparison with manually assigning attributes, which must be done otherwise.

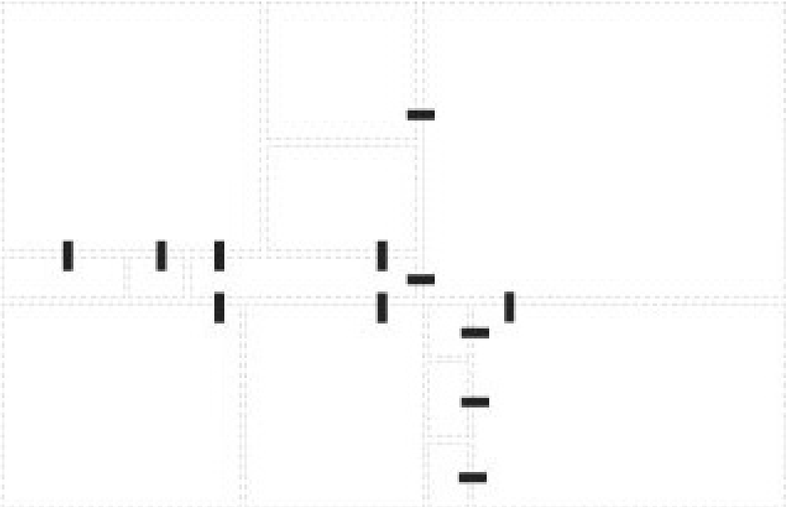
43 https://www.flexsim.com/jsp?arnumber=4786975, accessed 1 July 2017
44 https://www.autodesk.com/products/3ds-max/overview



a)

	Department 1	Department 1	Department 1
Department 1			
Department 1	x		
Department 1	y	x	
Department 1	z	x	y
Department 1	x	z	y
Department 1	y	z	x

b)



c)

Figure 32 Presets imported in NetLogo

a) Schema b) Table of frequencies c) Layer containing door openings

4.4.1. Schemata and Door Openings

In order to create an environment for a simulation one can import in a software package already prepared building plans or even to draw the plans directly in a simulation software, using present tools. One can import plans manually, in which case those are stored on computer's hard drive or the plans can be called from a database where are previously uploaded. Anyhow, the building plans are the essential part of a simulation. Without them, there is no built environment meaning agents do not have a 'world' to settle.

Figure 32a depicts a schema that consists of colored polygons. A process of creating a schema from fire escape plan is already described. One needs to keep in mind, that polygons should be separated by at least one pixel. If not so, during vectorization polygons will be united in one single room with one single color. Converting a schema to black and white image (Figure 18) will destroy room boundaries if separation is smaller than one pixel.

A legend containing department's colors and their color-coding should be written in form of a spreadsheet and exported as .csv file. In addition, a spreadsheet that contains frequencies between departments should also be pre-defined and ready to import in NetLogo. These data save us a time and reduce a manually labor because one spreadsheet that contains information on certain attributes replaces a hundreds of clicks in simulation software where we would have to set all attributes manually. Another useful layer contains door openings. As will be described later, the software package NetLogo has a certain logic that stands behind an agent's movement.

We can imagine a set of points placed in 3-dimensional orthogonal grid. Each point must be connected with the other with a line (link). Only so, an agent can move from point A to point B. Without a link, an agent cannot travel the distance between A and B.

If we observe the schema illustrated on Figure 32a, we notice a color-coded polygons separated by white areas which act as walls. As in the real world, an agent cannot go through the wall i.e. we do not create links that cross white areas. In order to provide realistic environment, one can create a layer door openings, which contains only lines that mark doors. Later on, one uses this layer in addition to schemata, creating a complete grid of links and waypoints, so an agent enters each room exactly at the location provided with the layer door openings.

4.4.2. Adjacency matrix

> Planning health care facilities usually begins with pre-defined room schedule as well as described relations between separate units.

Regardless of how the first planning stage is initiated (by competition or by direct contract), client needs to provide essential information on internal mechanisms of health care facility. In literature we can find numerous examples that illustrate those mechanisms (Figure 33) but due to the fact that every building has specific requirements, an input from contractor in terms of given spatial relations is fundamental. It is also quite common that contractor gives to disposal a conceptual schema that already suggests space arrangement in correct proportion.

Every written/drawn data that describes internal relations between organizational units must be translated into a language readable to NetLogo or any other program that we use in order to simulate an environment. Connection from one area to another can be described as mandatory, desirable or neutral. This words can be mapped to numbers [mandatory=1, desirable=0.5, neutral=0].[45]

45 A Case Study of Social Network Analysis in Adjacency Graphs, Gabriel Wurzer, Wolfgang E. Lorenz, 2016, in Proceeding of the 34th International Conference on Education and research in Computer Aided Architectural Design in Europe, At Oulu, Volume: 2, ISBN: 9789491207112, p. 229-238, http://www.academia.edu/28066107/SpaceBook_A_Case_Study_of_Social_Network_Analysis_in_Adjacency_Graphs, accessed 03 July 2017

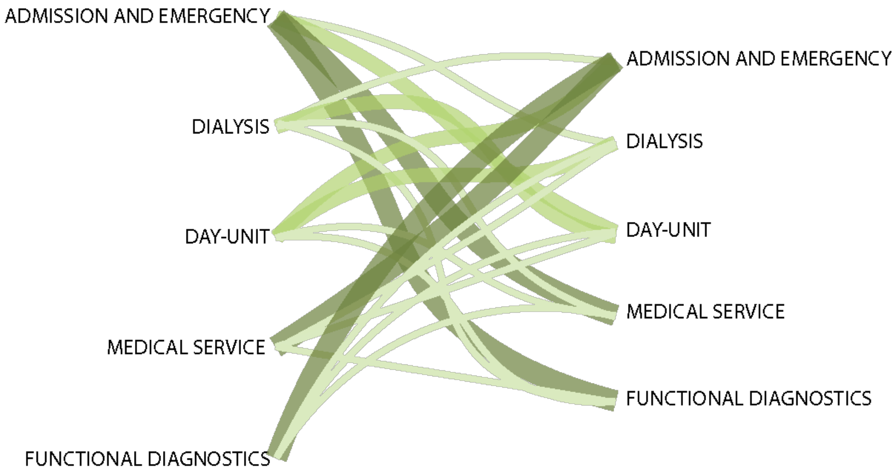


Figure 33 A line thickness depicts importance of the connectiona)

Relations between organizational units can also be described as close, distant and neutral.[46] The mentioned approaches refer to spatial organization. We observe in this sense two departments that must be close to each other or can be distant. Depending on how distant they are allowed to be i.e. how close they should be, we introduce a midpoint defined as neutral connection.

Closeness and distance are descriptive characteristics but they reveal simultaneously information on physical environment. However, they operate with functional demands, without introducing time and persons/agents as variables. If we want to show a personal/patient flow, we need to think about frequencies between departments rather than their spatial relations. Frequencies can be described as high, medium and low.

Here we meet a challenge. Although spatial arraignment of departments and people flow between them are closely related, traffic flow do not derive from spatial relations. For instance, two organizational units are planned (with intention!) next to each other what would mark this relation as direct, mandatory or close. It can be connection between maternity and emergency department. From pure functional reasons, this connection should be good, but it does not mean that patient flow between these organizational units is intensive. Our problem lies in a fact that it is not easy to discover real frequencies of people flow between departments.

Information that we as planners get from contractor or find in

46 Adjacency in hospital planning, Wolfgang E. Lorenz, Martin Bicher, Gabriel X. Wurzer, 2015, IFAC-PapersOnLine, ISSN: 1474-667, p.862-867, https://www.semanticscholar.org/paper/Adjacency-in-hospital-planning-Lorenz-Bicher/303864b5149247c17fa3dd08f56987f4173c0c51, accessed 03 July 2017

literature refer to spatial arrangement. Starting with the most famous architectural manual, Architect's Data from Ernst Neufert, information on internal mechanisms of hospitals are given through adjacency matrix that describes spatial connections and not people flow.

This is very logical because people flow depends on the size of a hospital, region, total number of patient etc. Only way to create adjacency matrix with correct information would be to get these from HIS (Hospital Information System). This is possible in the case of renovation or if we are planning the new building as the replacement for the old one, so we can get all necessary information from the HIS of the old hospital. This is also only one part of the solution. If we even get the information, the question is how much time we should invest in analyzing and structuring the data.

In this work, we will only assume approximate number of patients and staff moving between departments in order to depict systematically all functional areas and their interaction. Matrix could be designed in a manner that we consider every interaction between two departments as reciprocal. It means that personal/patient flow from area A to B is exact the same as the one from B to A. That can be true but not necessarily. Still, we will simplify the matrix and pretend that this reciprocity is always the case.

After we put all hospital departments in a spreadsheet and assign them correspondingly a number [1, 0.5 or 0], we export the table as e.g. .csv file which is appropriate for further work with NetLogo. On that basis, three main lists will be created, each for a specific frequency level (low, medium, or high frequency). One list contains sublists i.e. pairs of departments shown in adjacency matrix (Figure 34).

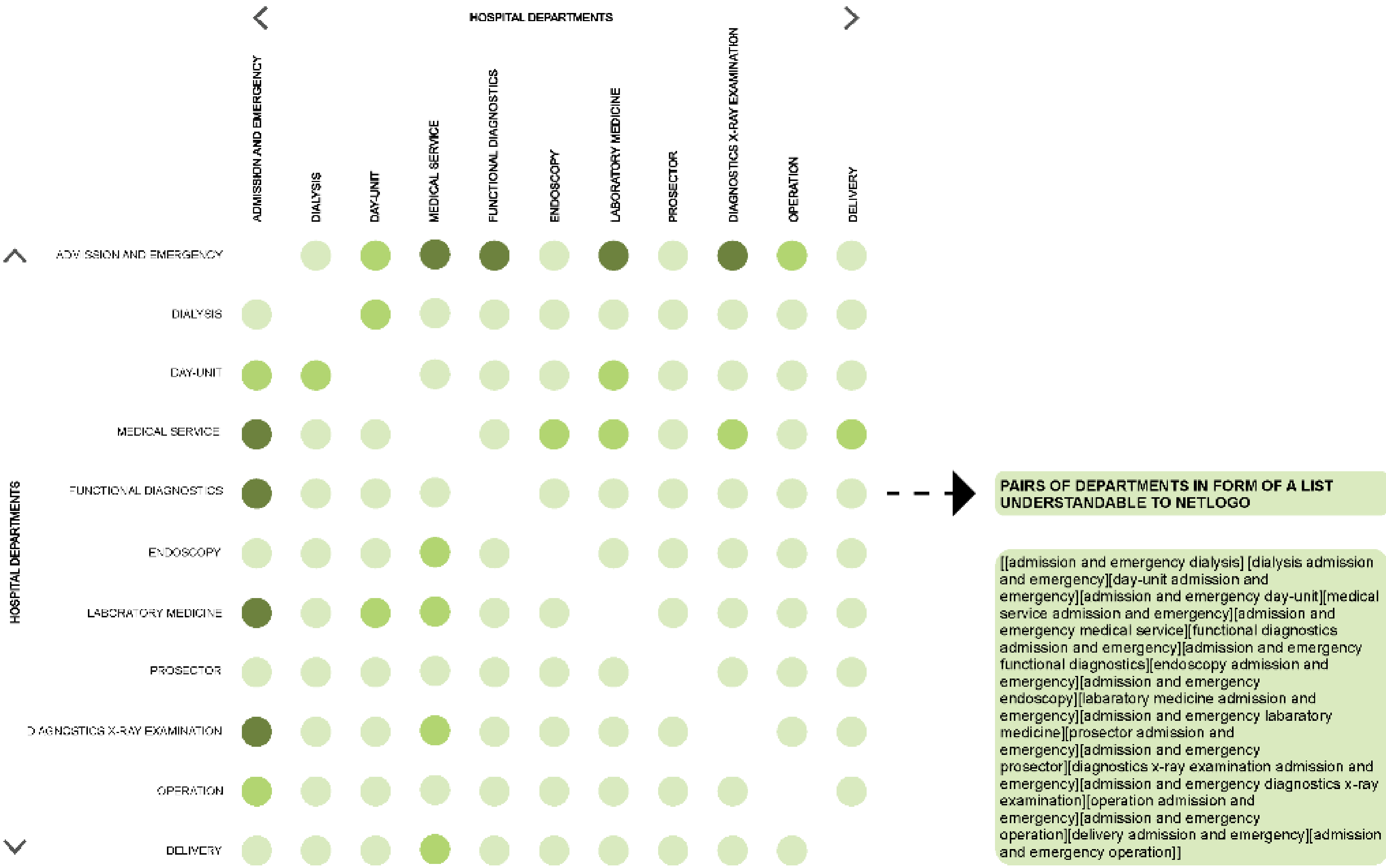


Figure 34 Adjacency matrix given with a spreadsheet, exported in proper format which NetLogo reads as lists

4.5. Simulation Software

As outlined above, the simulation research is conducted in NetLogo. It is user-friendly ABM software authored by Uri Wilensky 1999. It was originally developed at the Center for Connected Learning and Computer-Based Modeling at Northwestern University in an attempt to run StarLogo on Mac OS.[47]

Since non-programmer friendly, it is quite adoptable for architects due to their modest knowledge about computer science. Both 2D and 3D environments are available though NetLogo's support for 3D is less developed than the one for NetLogo 2D.

On its homepage, one can find extensive documentation and tutorials. NetLogo supports GIS, can record movies of simulation and comes with a large library of sample models. Beside the built-in library, there are NetLogo user community models, free to download and use. User manual is additionally translated in Chinese, Czech, Japanese and Spanish.

The NetLogo world exist on the Cartesian coordinate system. The center of the coordinate system (0, 0) is the origin located in the physical center of the NetLogo world, which is composed of grid cells. Each patch is defined by the coordinate at its center.[48]

Simulation environment consists from turtles, patches, links, and the observer[49] (Figure 35). Brief description of these main components is given with the Figure 36. Turtles and patches are agents with the ability to interact with other agents. The main difference between patches and turtles is the fact that patches cannot move. Patches have always integer coordinates, while turtles can have decimal coordinates.

Turtles are usually used to represent the main research target. Patches on the contrary serve as a static ground over which turtles move. A turtle can take shape of a person, car, tree, sheep or any desirable shape that one can create directly in NetLogo.

47 Agent-based modeling in urban and architectural research: A brief literature review, Liang Chen, 2012, Frontiers of Architectural Research Volume 1, Issue 2, ISBN: p.166-177, http://www.sciencedirect.com/science/article/pii/S2095263512000167?via%3Dihub, access July 2017

48 https://publik.tuwien.ac.at/files/PubDat_200771.pdf
49 http://ccl.northwestern.edu/netlogo/docs/

Links are connections between turtles. Turtles, patches and links have their own built-in variables (Figure 37) which define position, color, size and many other attributes.

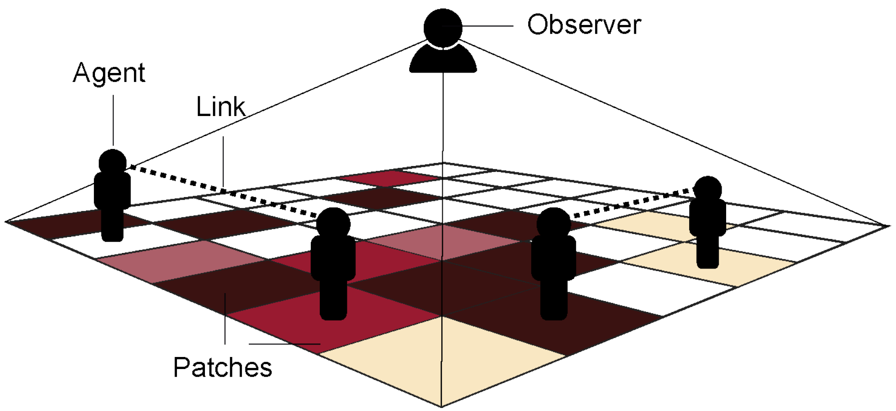


Figure 35 Simulation environment in NetLogo

Item	Function
Patches	The ground over which the turtles move.
Turtles	Agents that move around in the simulation world.
Links	Connections between turtles
Observer	The invisible control center that oversees the world and gives a command to the patches or turtles.

Figure 36 Main components in NetLogo

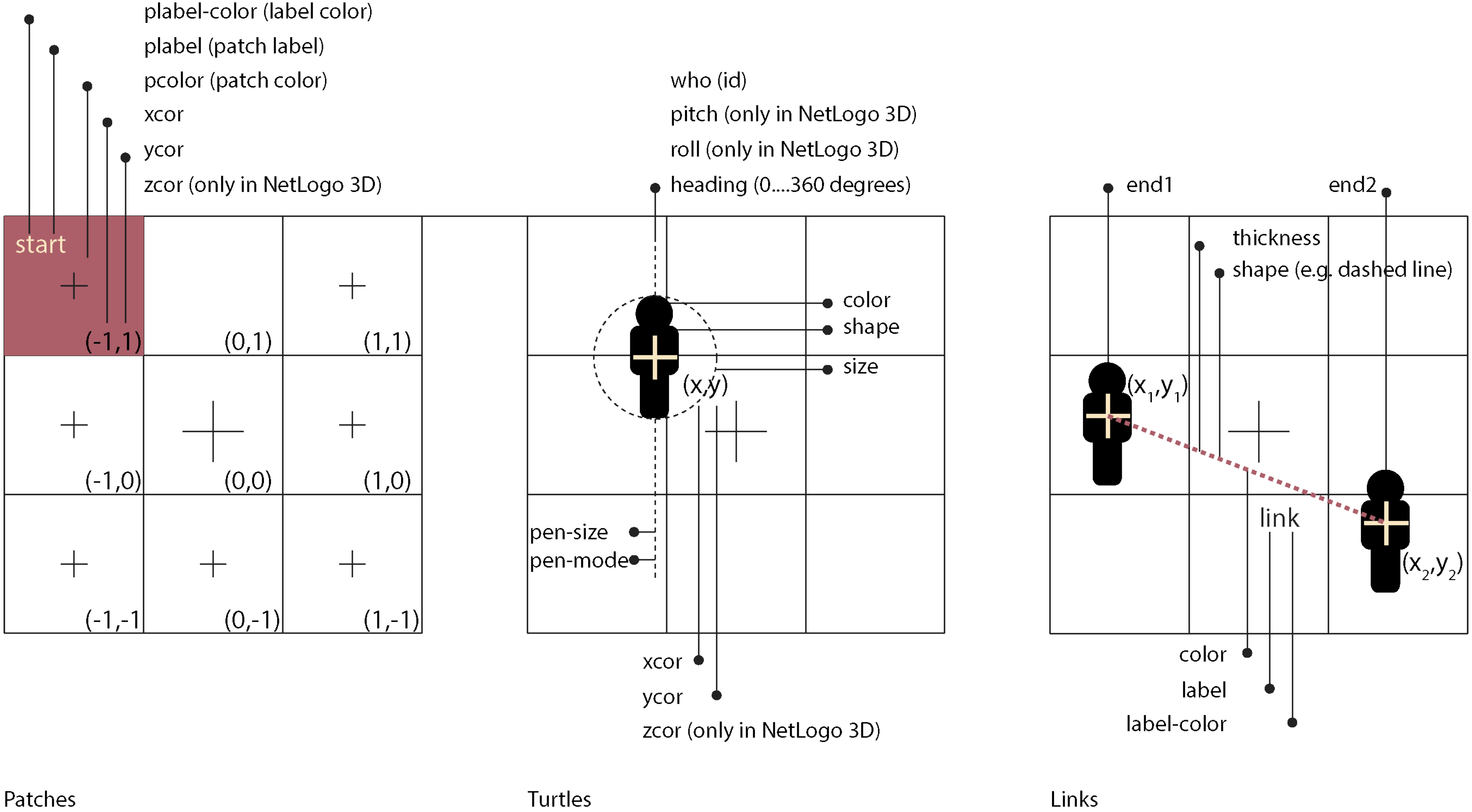


Figure 37 Built-in variables in NetLogo

1.

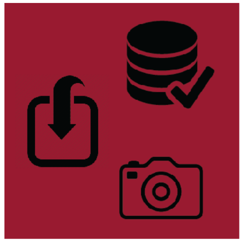
Introduction

2.

Computer Simulation as a Planning Tool for Architect

3.

Related Work



4.

Improved Workflow

5.

Case Study Baden

6.

Discussion

5. Case Study Baden

In order to contextualize conducted research, a concrete example will be presented in the following chapter.

Landesklinikum Baden (LKH) is built in 1885 as Rath'sche Badner Krankenhaus with the capacity of 36 beds. In 1897, the building was extensively expanded. Over decades, the building was renovated, expanded and renamed a few times. A new hospital building is opened in 1983, with its 2 characteristic polygonal towers and 352 beds. Since 2012, three new pavilions are being built, close by the old hospital, which is demolished.

Each pavilion has three upper floors and maximal heights from 16m. The net area is about 30,980 m² and the gross floor area 61,590 m².

Departments included in the new building are depicted on the Figure 38. LKH Baden is a good example how a simulation could be used as a planning tool. Namely, during numerous renovations and reconstructions, simulation could help planners to control their new designs much better, whereby input data, such as the exact number of patients, can be taken from the HIS of an old facility. The construction works on the new building started in 2010 and for the next 6 years, the old hospital was operating next to the new one. Accordingly, a simulation based on authentic information could show a weaknesses and strengths in every phase of the new project.

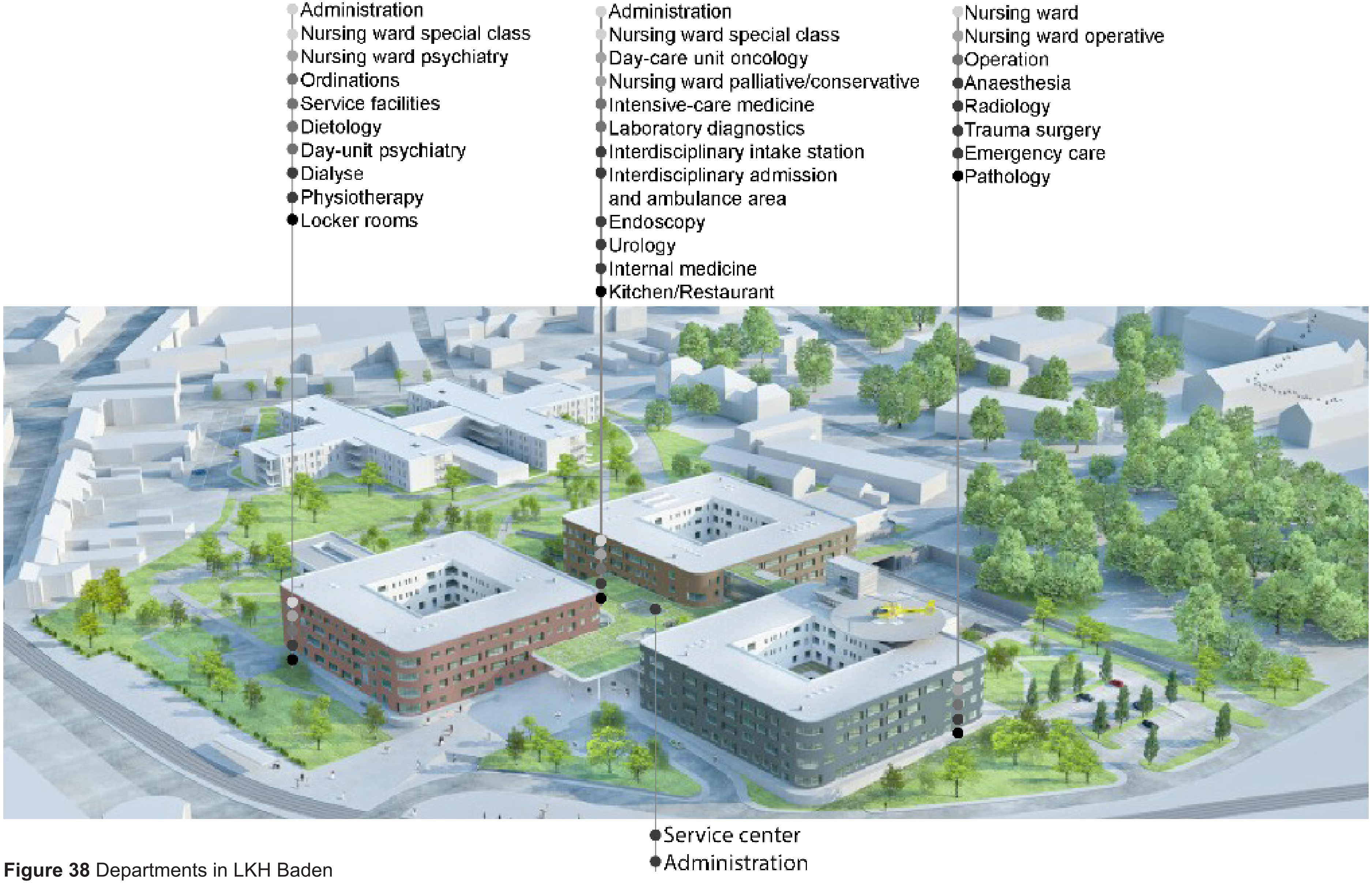


Figure 38 Departments in LKH Baden

5.1. Milestones

The Figure 39 illustrates key milestones that we need to cross from the very beginning to the completion of the simulation. The workflow shown previously in the Figure 7 is shortened for a phase of adjusting i.e. stitching plans together, since we manage to get plans directly from the architecture office.

After preparing necessary data for simulation, we had a choice: a) to import data directly in NetLogo and b) to use benefits of DB instead. Test version of the project is conducted without DB but at one point, we switched to DB, since it was much easier to manage and coordinate the assignment

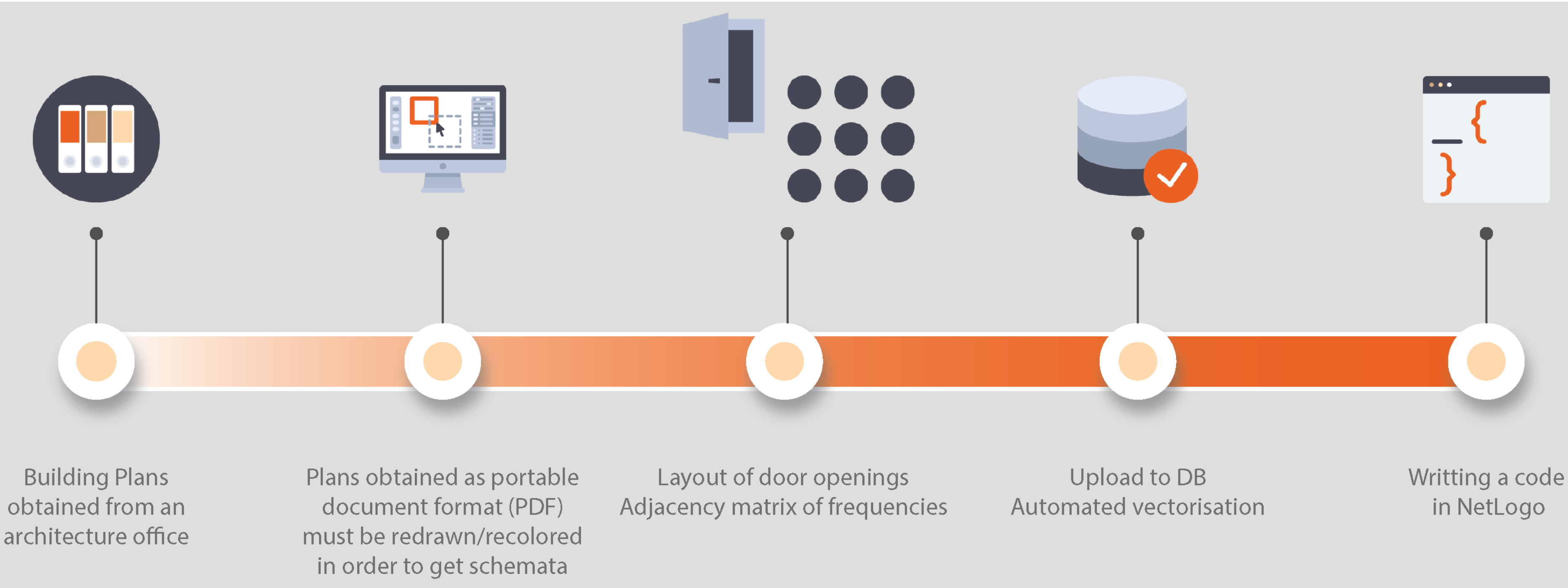


Figure 39 Case Study Baden: Milestones

5.2. Obtaining building plans and their further editing

As already mentioned, building plans could be found in various ways. Photographing fire escape plans is described as the most practical and fastest way to achieve aforementioned. Nonetheless, we succeeded to obtain Baden project directly from architecture office that designed the building. It saved us a certain time that would be spent in hospital visiting and photographing fire escape plans. As we got the hospital plans as portable document format (PDF), further adjustments were necessary.

At this point, we could a) use Photoshop or any other drawing package with the flood-fill tool or b) use any CAD application or any raster graphics editor in order to re-draw the plans. The both options are equally simple and take approximately the same amount of time.

With CAD one needs to recreate every room using a filling tool. Using a Photoshop, one can employ only a paint bucket, whereby the result is a bit poor (Figure 14a).

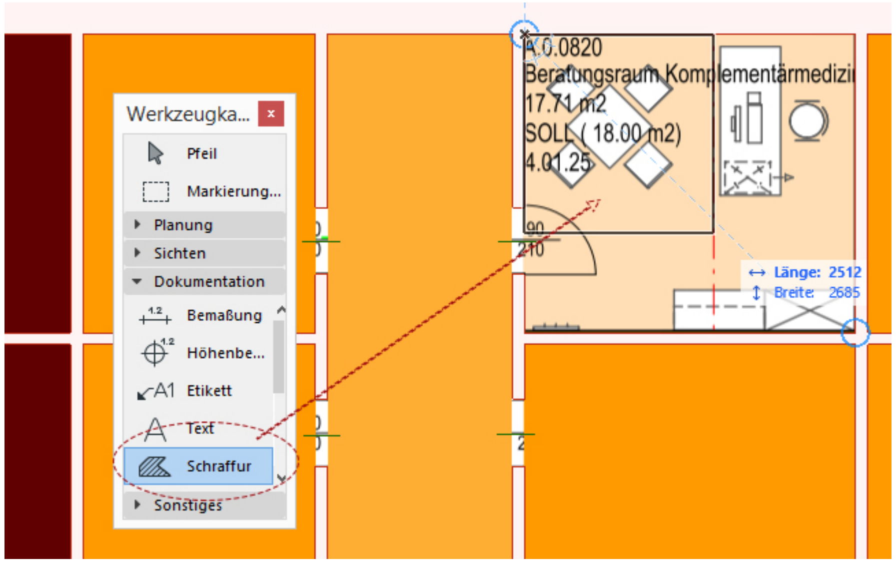


Figure 40 Polygons redrawn in Photoshop

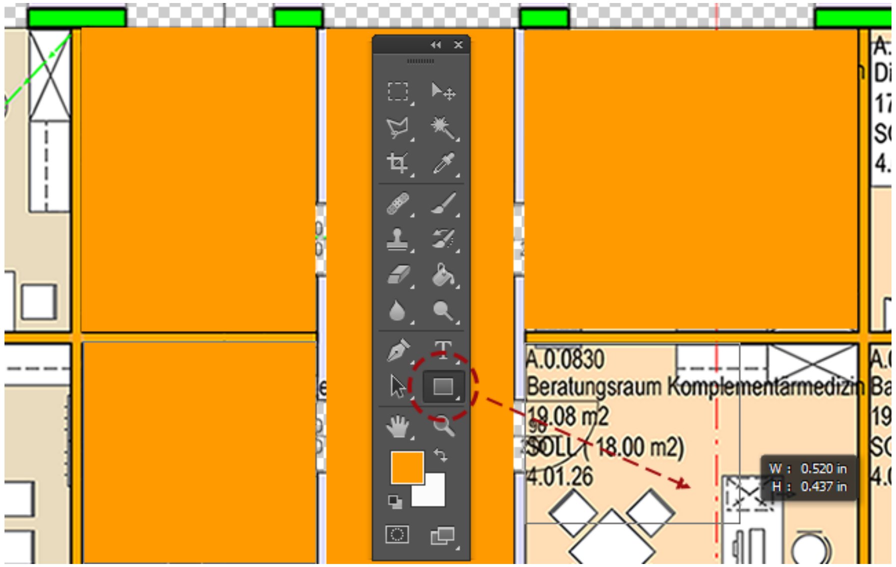


Figure 41 Polygons redrawn in Archicad

Instead, one can also use rectangle tool in order to redraw polygons (Figure 40).

Paint bucket tool is rather inappropriate when one works with complex floor plans with many details. Paint bucket algorithm always looks for closed surfaces, which is many, considering furniture, dimensions etc. (Figure 42).

In case of LKH Baden there is more than 400 polygons in one pavilion, which is why every extra mouse click in one room should be repeated over thousand times. Accordingly, it is irrelevant which software one use as long as he is able to recreate all surface areas in form of colored polygons reducing a number of operations to a minimum. Assignment is completed, however, in Archicad, since it was the easiest solution to redraw polygons from scratch than to clean up all unnecessary details (Figure 41).

Figure 43 illustrates the result i.e. the schema.

(Door openings layout) Besides schemata, we will use for the simulation a door openings layout described in the Chapter 4.4.1. Considering two main groups of users, patients and staff, two separate layouts for each floor are developed. One layout contains only door openings, which are used by patients and other one contains all of them. That way, staff can enter every room and patients cannot enter staff only areas.

The layout consists of lines, each positioned exactly in the middle of a specific door symbol. As well as schemata, created layouts will be saved as bitmaps in small resolution, uploaded to DB and vectorized accordingly. Later, a code that we will write in NetLogo will ask for all these images. Figure 44 illustrates the schemata and the layouts of the door openings, for staff and patients retrospectively.

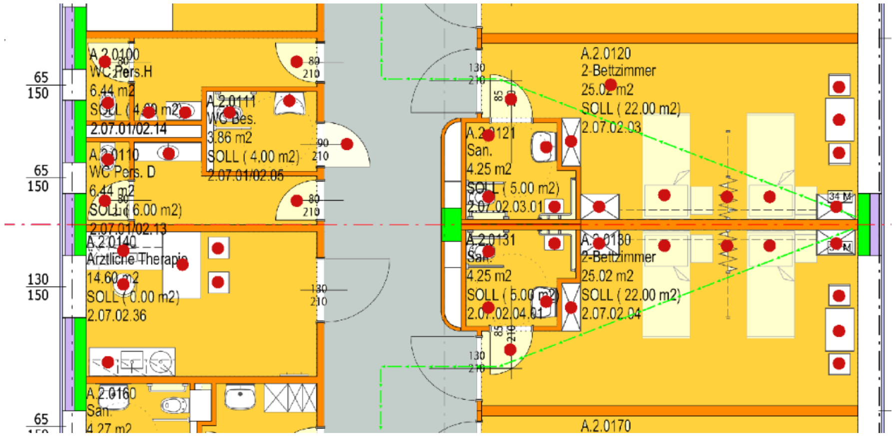


Figure 42 LKH Baden, floor plan segment edited in Photoshop by using flood filling only.
Since furniture, door openings and room labels (highlighted with red dots) impede a finer result, it is better to clean up the plans ahead or to perform the operation by using a marquee tool instead. In that manner problem comes to re-drawing, so it is irrelevant if CAD application or Photoshop (or similar) will be used.

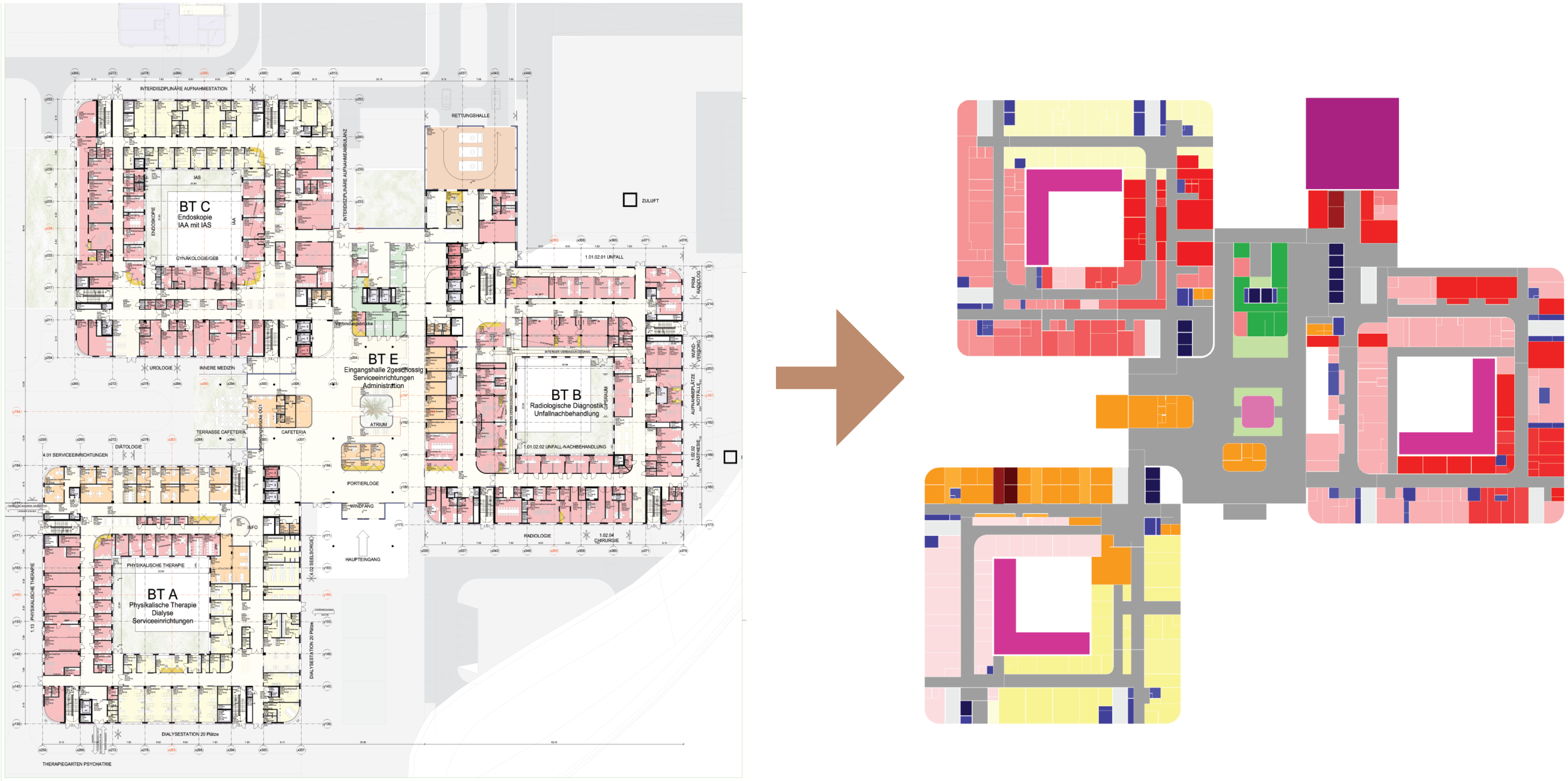


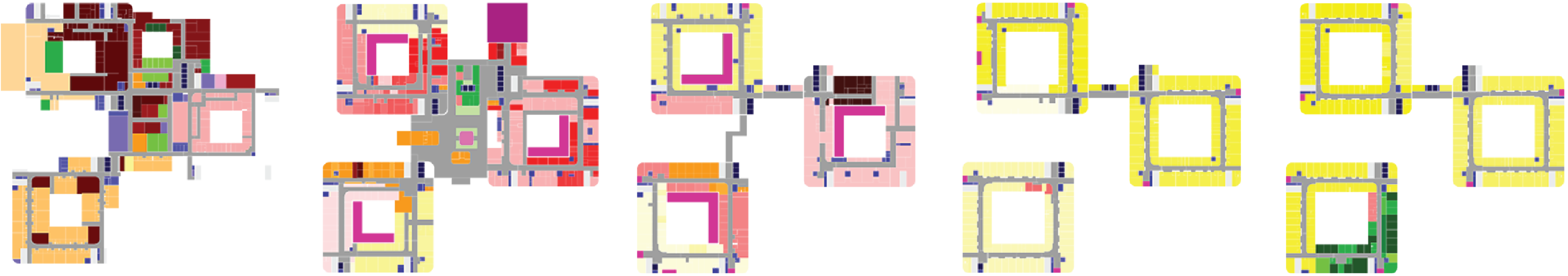
Figure 43 Transition between original floor plan and schema

(Adjacency Matrix) Before schemata have been reproduced, appropriate legend is established. Schemata contain 89 colors, which represent functional areas of the hospital. The colors and corresponding areas are given with the legend on the Figure 45.

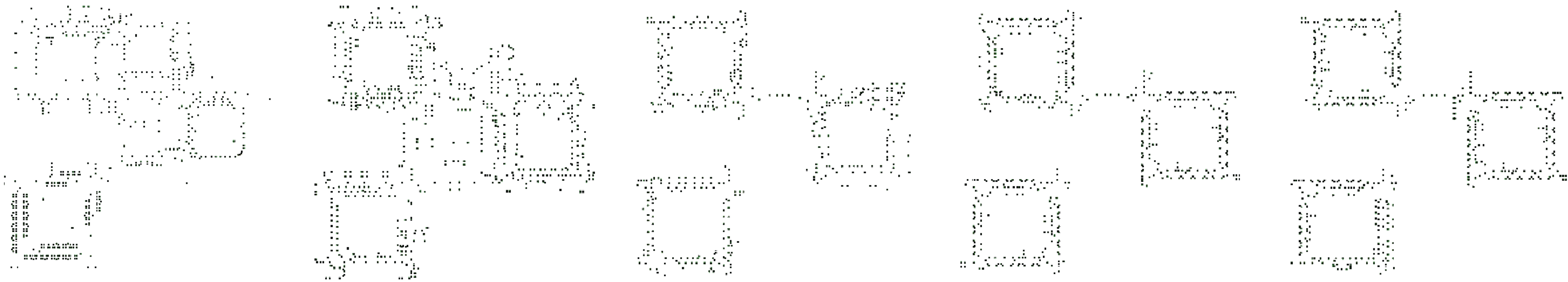
This legend needs to be translated into a spreadsheet in a form of adjacency matrix. By means of simple math, one can calculate the total number of frequencies between functional areas. In case of 89 functional areas this number is 89*89-89 (minus 89 because we neglect the connection of a department with itself) which is 7832. So many outcomes lead to unnecessary overload of NetLogo so it was a logical step to reduce a number of departments in a table of frequencies. Instead of 89 functional areas, we decided to work with 31.

It is done in a way that similar functional areas are simply covered with one name in the table of frequencies (Figure 47). Using the same logic, 31 functional area is listed in the spreadsheet that contains the departments and corresponding colors (Figure 45).

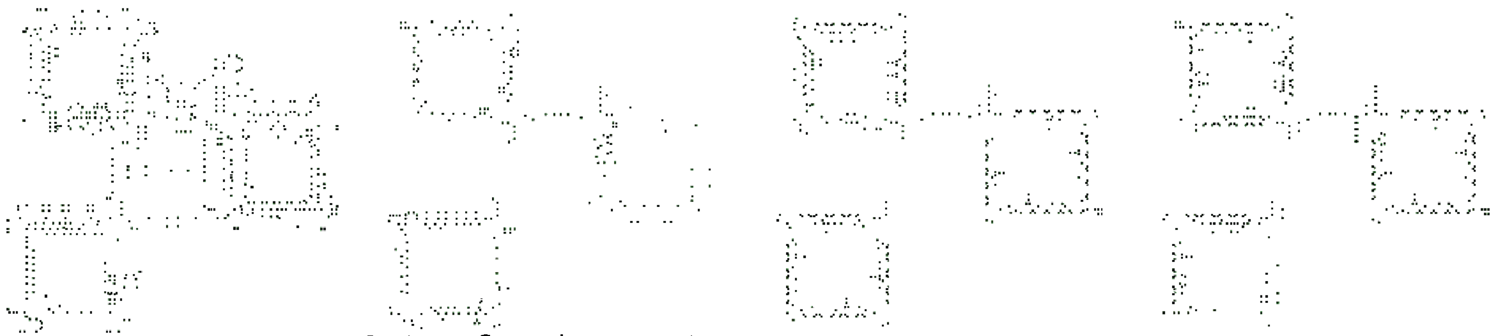
Although it leads to a certain mismatch, knowing that schemata contain 89 and spreadsheets almost three times fewer colors, the input data fulfills the task. We chose a treatment chains (pathways of patients) that will be simulated in a way that aforementioned mismatch do not impedes the task. This problem could be solved by uploading updated schemata to the DB, but those would provide less information for any further use.



Schemata: colored polygons stand for funktional areas



Staff flow: door openings



Patients flow: door openings

Figure 44 Layers imported in NetLogo

1. Treatment and Examination

1.01 Admission and Emergency Care
1.01.01 Interdisciplinary Intake Station
1.01.01.a IIS WL
1.01.01.b IIS WS
1.01.02. Accidents Initial Examination and post-treatments
1.01.02.a Accidents WL
1.01.02.b Accidents WS

1.02 Medical Services
1.02.02 Anaesthesia
1.02.02.a Anaesthesia WL
1.02.02.b Anaesthesia WS

1.02.04 Surgery
1.02.04.a Surgery WL
1.02.04.b Surgery WS

1.02.06 Gynecology and Obstetrics
1.02.06.a Gynecology and Obstetrics WL
1.02.06.b Gynecology and Obstetrics WS

1.02.10 Internal Medicine
1.02.10.a Internal Medicine WL
1.02.10.b Internal Medicine WS

1.02.20 Emergency Psychiatry

1.02.22 Urology
1.02.22.a Urology WL
1.02.22.a Urology WS

1.02.23 Central Roms for the Medical Service

1.03 Functional Diagnostics
1.03.a Functional Diagnostics WS

1.04 Endoscopy
1.04.a Endoscopy WL
1.04.b Endoscopy WS

1.05 Laboratory Medicine

1.06 Pathology

1.07 Diagnostic X-ray Examination

1.09 Operation

1.10 Delivery
1.10.a Delivery WL
1.10.b Delivery WS

1.13 Physiotherapy
1.13.a Physiotherapy WL
1.13.b Physiotherapy WS

2. Care

2.01 General Care
2.01.01 General Care - Nursing Station 1

2.01.02 General Care - Nursing Station 2
2.01.02.a General Care WS

2.01.03 General Care - Nursing Station 3

2.01.04 Internal Medicine/Special Class

2.01.05 Operational I - Nursing Station 1

2.01.06 Operational I - Nursing Station 2

2.01.07 Operational II Acc./Surg./Uro.- Nursing Station 1

2.01.08 Operational II Acc./Surg./Uro.- Nursing Station 2

2.01.09 Special Class Operational

2.03 Intensive Care Medicine
2.03.a Intensive Care Medicine WL

2.04 Dialyse
2.04.a Dialyse WS

2.07 Nursing Station - Psychic Illnesses
2.07.01 Nursing Station 1 - Psychic Illnesses
2.07.02 Nursing Station 2 - Psychic Illnesses

2.09 Addmision - Nursing Care
2.09 .a Addmision Nursing Care WS

2.11 Day Unit
2.11.01 Day Unit Psychiatry
2.11.03 Day Unit Oncology

3. Administration

3.01 Management and Administartion
3.01.01 Management
3.01.01.a Management WS

3.01.02 Administration
3.01.02.a Administration WS
3.01.02.b Administration WL

3.02 Archiving

3.03 Information and Documentation

3.04 Library

4. Sozial Services

4.01 Service Facilities
4.1.a Service Facilities WS

4.03 Locker Rooms

4.04 Food Supply

Figure 45 The color-coded legend of LKH Baden

5. Disposal and Supply Management

- 5.1 Supply of Pharmaceuticals
- 5.2 Sterile Supply
- 5.5 Food Supply
- 5.6 Laundry Service
- 5.7 Warehousing and transfer of goods
- 5.8 Maintenance
- 5.9 Waste Menagement
- 5.10 Transport

7. Miscellaneous

- 7.00.02 Ambulance Entrance
- 7.00.08 Loggias, Balconies
- 7.00.09 Terraces, Roofgarden
- 7.00.10 Atria, Garden
- 7.00.13 Loading Yard

8. Technic

- 8.00.01 Elevators
- 8.00.02 Shafts
- 8.00.03 IT
- 8.00.04 Plant rooms

9. Traffic

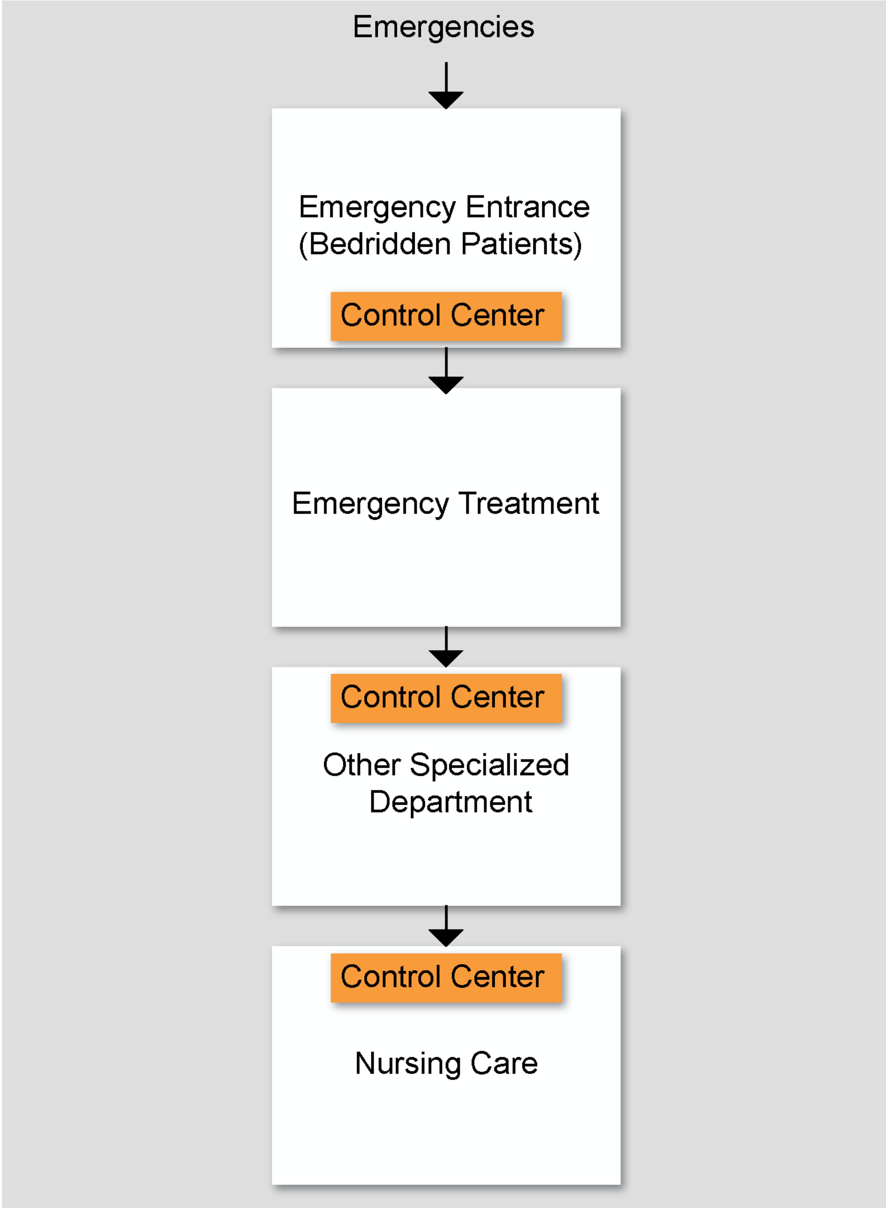
- 9.00.01 Stairs
- 9.00.02 Hallways

Color	Hospital Department
#300b09	Admission and Emergency Care
#5a1511	Medical Services
#851e19	Functional Diagnostics
#af2820	Endoscopy
#d73229	Laboratory Medicine
#df5b54	Pathology
#e7847f	Diagnostic X-ray Examination
#efada9	Operation
#f7d6d4	Delivery
#f2ccd5	Physiotherapy
#35350b	General Care
#636314	Intensive Care Medicine
#92921d	Dialyse
#c1c127	Nursing Station - Psychic Illnesses
#eded31	Addmision - Nursing Care
#f0f05a	Day Unit
#125818	Management and Administartion
#1a8124	Archiving
#23ab2f	Information and Documentation
#2cd13b	Library
#652c08	Service Facilities
#95410c	Locker Rooms
#c55610	Food Supply
#231810	Supply of Pharmaceuticals
#422e1e	Sterile Supply
#61442c	Food Supply
#7f5939	Laundry Service
#9d6e48	Warehousing and transfer of goods
#b08b6c	Maintenance
#c4a891	Waste Menagement
#d8c5b6	Transport

	Admission and Emergency Care	Medical Services	Functional Diagnostics	Endoscopy	Laboratory Medicine	Pathology	Diagnostic X-ray Examination	Operation	Delivery	Physiotherapy	General Care	Intensive Care Medicine	Dialyse	Nursing Station - Psychic Illnesses	Addmision - Nursing Care	Day Unit	Management and Administartion	Archiving	Information and Documentation	Library	Service Facilities	Locker Rooms	Food Supply	Supply of Pharmaceuticals	Sterile Supply	Food Supply	Laundry Service	Warehousing and transfer of goods	Maintenance	Waste Menagement	Transport
Admission and Emergency Care																															
Medical Services	1																														
Functional Diagnostics	1	1																													
Endoscopy	0.5	0.5	0.5																												
Laboratory Medicine	1	1	0.5	0																											
Pathology	0.5	0	0	0	0																										
Diagnostic X-ray Examination	1	1	1	0	0	0																									
Operation	1	1	1	0.5	0.5	0	0.5																								
Delivery	1	1	0	0	0	0	0	1																							
Physiotherapy	0.5	0.5	0	0	0	0	1	0	0																						
General Care	1	1	0.5	0.5	0	0	0.5	1	1	1																					
Intensive Care Medicine	0	0	1	0	0	0	0	1	0	0	0																				
Dialyse	1	1	1	0	1	0	0	0	0	0	0	0																			
Nursing Station - Psychic Illnesses	0.5	0.5	0	0	0	0	0	0	0	0	0	0	0																		
Addmision Nursing Care	1	0.5	0.5	0.5	0.5	0	0.5	1	1	0	1	1	0.5	0																	
Day Unit	1	1	1	1	1	0	1	1	0	0	0	0	0	0	0																
Management and Administartion	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0															
Archiving	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1														
Information and Documentation	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1													
Library	0	0	0	0	0	0	0	0	0	0	0.5	0	0	0.5	0	0	0.5	0	0												
Service Facilities	0	0	0	0	0	0	0	0	0	0	0.5	0	0	0.5	0	0	0	0	0	0											
Locker Rooms	0.5	0.5	0.5	0.5	0.5	0	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0	0	0	0	0										
Food Supply	0.5	0.5	0.5	0.5	0.5	0	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0	0	0	0.5	0									
Supply of Pharmaceuticals	0.5	0.5	0.5	0.5	0.5	0	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0	0	0	0	0	0	0								
Sterile Supply	0.5	0.5	0.5	0.5	0.5	0	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0	0	0	0	0	0	0	1							
Food Supply	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	1	1	0	0						
Laundry Service	0.5	0.5	0.5	0.5	0.5	0	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	1	0.5	0	0.5	0					
Warehousing and transfer of goods	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0				
Maintenance	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
Waste Menagement	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
Transport	0	0	0	0	0	0	0	0	0	0	0.5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Figure 47 Table of frequencies

5.3. Treatment chains



Simulation is implemented for two different scenarios.

- Pathway of a patient admitted in the emergency department
- Pathway of a patien admitted to outpatient treatment and examination

Figure 48 Pathway of a patient admitted in the emergency department

5.3.1. Pathway of a patient admitted in the emergency department

In general, emergency patients reach hospital trough emergency entrance. The patient is treated in examination and treatment chambers, until transfer to spezialized departmets, operating theater and afterward to general i.e. intensice care. Contact between examination and treatment departments and nursing ward should ocure trough control points of each department.[50] Control points are provided with information on each patient through HIS.

Figure 48 illustrates a general case of a pathway for a patient admitted to emergency department. In order to contextualize a given example, Figure 49 depicts a concrete situation. This scenario foresees a patient admitted to emergency department with multiple injuries after an accident.

A task-oriented care plan begins with triage, where a degree of urgency needs to be established. In Trauma Room, an initial care of severely injured person is provided. Depending on clinical picture, patient in forwarded to X-Ray, Ultrasound or CT if suspected head/brain/spine injuries or severe injuries of internal organs. After completed diagnostics, whereby all injuries are located and established, the patient is transferred to operating theater. Before operation, he will be prepared in a room for anesthesia.

Stationary patient i.e. patient scheduled for an operation usually spent time in a preoperative holding area, which provide the environment for calming, informative interactions that should help patients prepare for their surgical procedures.

After the surgury is completed, the patient is moved to recovery room, which is usually intensiv care. When the patient is stabil, he is transferred to general care. His further treatment and examinations can occure during his retention in hospital (as an inpatient) but most probably he will only visit a hospital if necessary for his rehabilitation as an outpatinet.

50 Das Friesen-Konzept für das Krankenhaus und für das Gesundheitswesen von morgen, Robert Wischer, Hille Rau, Karl Krämer Verlag, 1988, ISBN: 3782840011, p.90

51 Responsibilities of the preoperative holding area nurse, Dunn D., AORN Journal Volume 66, Issue 5, 1997, http://www.aornjournal.org/article/S0001-2092(06)62663-X/pdf, p.820 , accessed 02.07.2017

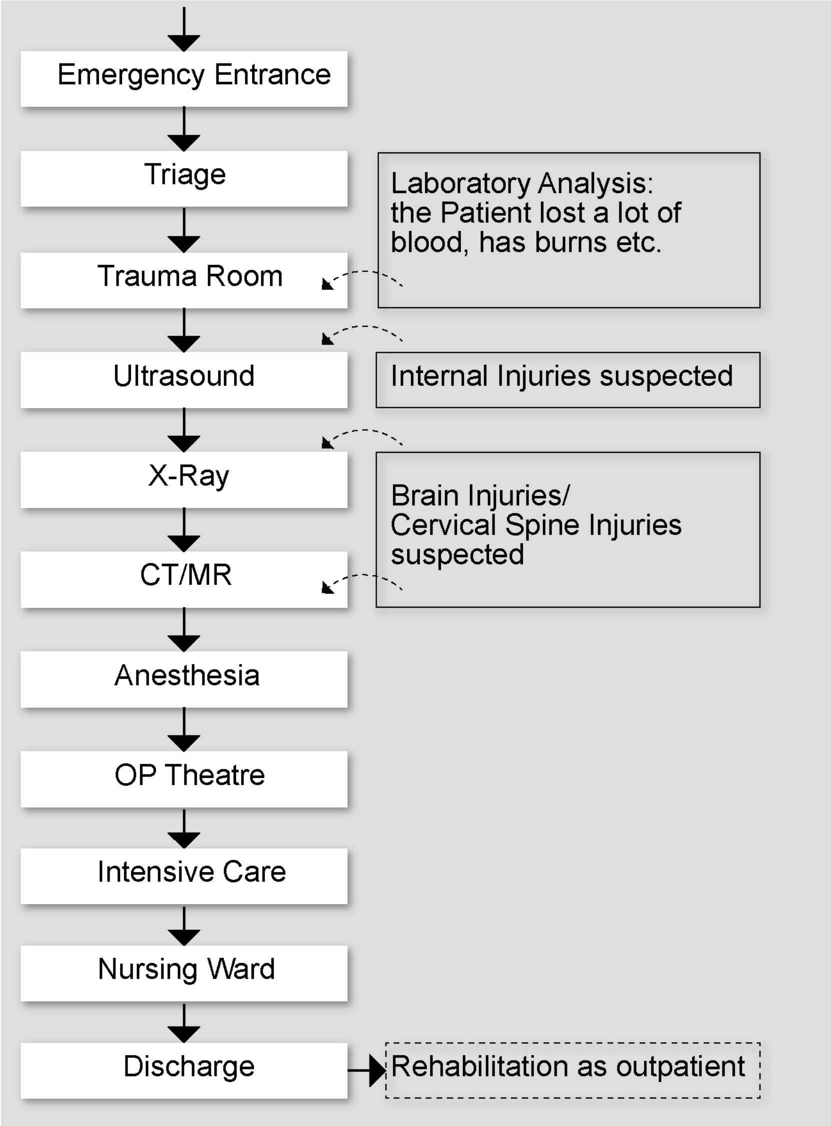


Figure 49 Assessment in the Emergency Department for Patients with Multiple Injuries after Car Accident

5.3.2. Pathway of a patient admitted to outpatient´s treatment and examination

Outpatients can be classified into several categories, depending on how they were referred to hospital.[52] Figure 50 illustrates a standard procedure that varies depending on previous mentioned categories.

A patient who is first time in the hospital receives information on info point and proceeds to central admission office. Depending on reason of visit, he will be forwarded to specialized department. If a patient comes without any referral from a doctor, first he needs to be examined from a front line clinician and then if necessarly forwarded to further examinations.

Outpatients, which come to a hospital for second time or more, already know in most cases where should they go.

If they do not have an appointment, admission and further instructions will be given at central admission office. If they do

52 Das Friesen-Konzept für das Krankenhaus und für das Gesundheitswesen von morgen, Robert Wischer, Hille Rau, Karl Krämer Verlag, 1988, ISBN: 3782840011, p.90 Patients with or without referral; patients with or without appointment; patients first time admitted to hospital or come back.

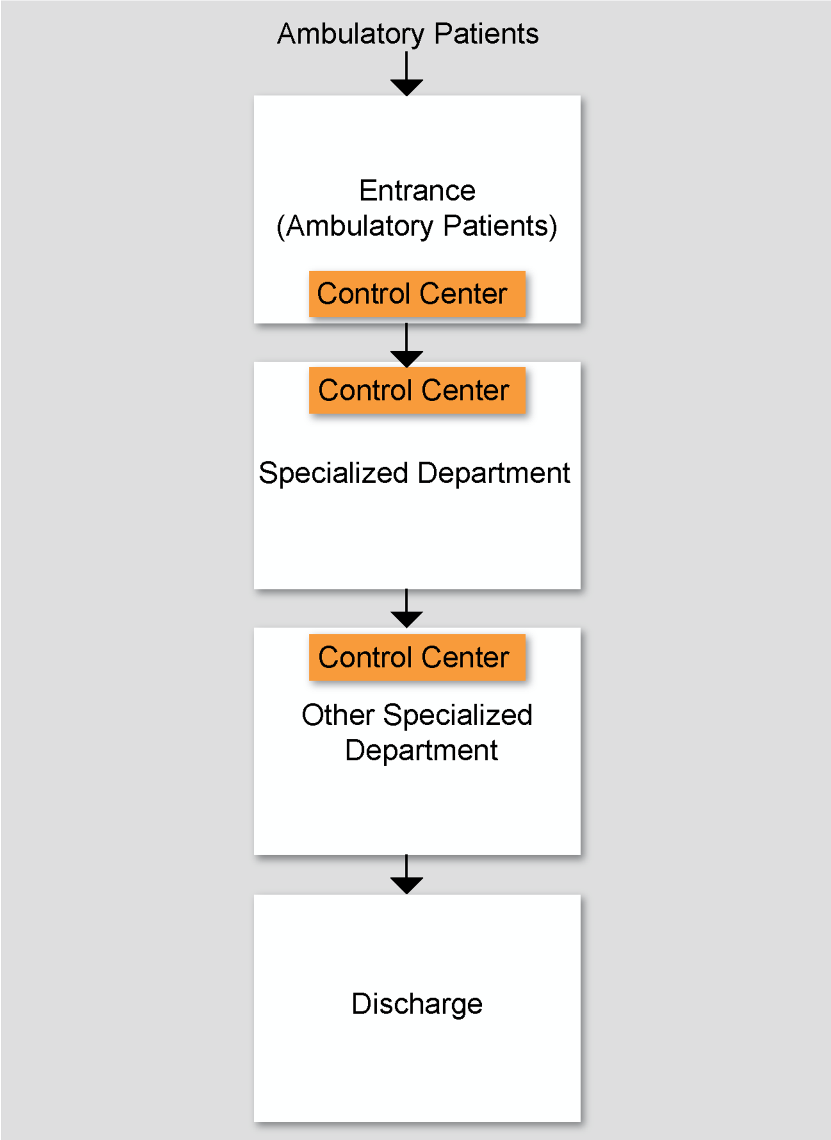


Figure 50 Pathway of a patient admitted to outpatient´s treatment and examination

have an appointment, they can check-in directly at the control center of the respective department.

Figure 51 illustrate a concrete example: a pathway of an outpatient with a medical referral to endoscopy.As the patient is first time in hospital, his route starts with information desk. He is forwarded to central admission, which perform initial registration in HIS and gives him instructions/directions to the control point of endoscopy.

After i.e. before check-in in respective department, a certain waiting time may occur. The next step of the pathway is preparation for the examination and endoscopy afterwards. Patient´s return occurs over department´s or central control point depending on circumstances and further examinations or treatments if necessary.

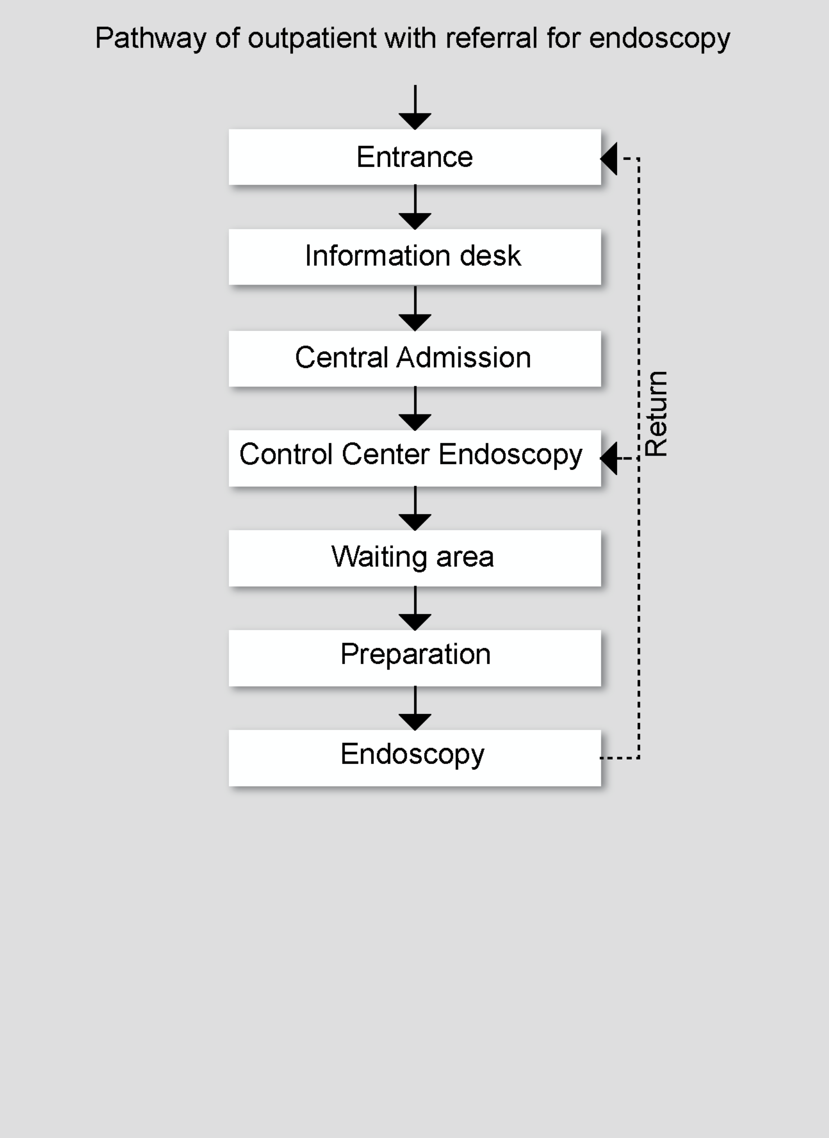


Figure 51 Pathway of a patient from information desk to endoscopy

5.4. NetLogo User Interface

Figure 52 illustrates NetLogo user interface. The software enables one to create buttons, choosers, switchers, sliders etc. First input field named serviceurl gives a web address of the database where all input data have been stored. Chooser building gives an opportunity to work with different buildings simultaneously inside of one model. The next chooser display-what offers four possibilities:

- none
- raster
- circulation
- geometry

The option raster depicts the building in form of blocks (turtles) These colorful cubes are waypoints and linking them one gets a grid depicted with circulation, a ground base on which an agent can move.

The logic behind this will be explained in the next chapter

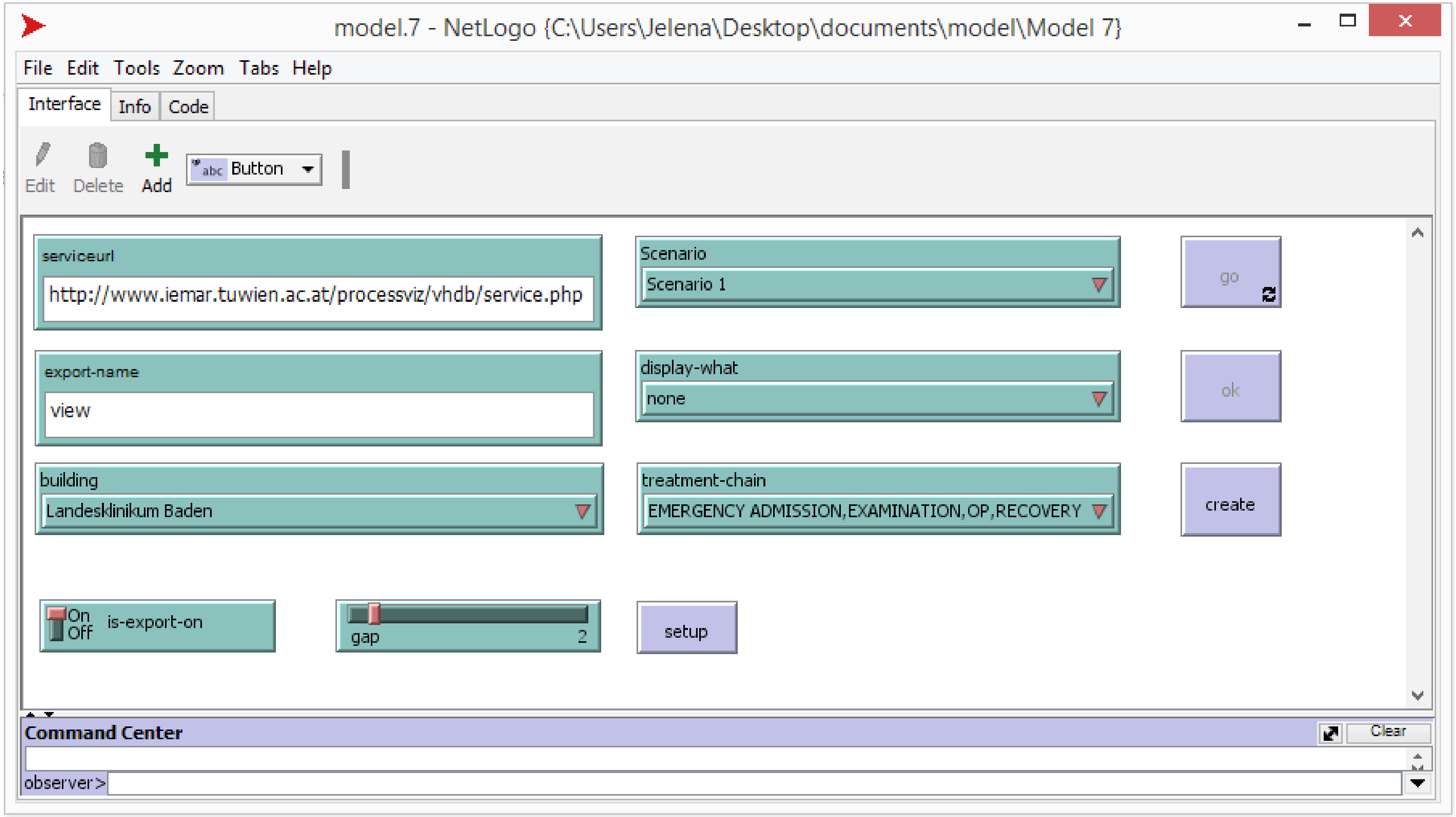


Figure 52 NetLogo User Interface

although is worth to mention that the grid we see by choosing the circulation is a requirement for movement of agents. The last option, geometry, is concerned with graphical style of the model. It shows the building not trough turtles (cubes) but rather trough tiny lines (edges) that represent the walls. Instead of unrealistic turtle-related appearance, by choosing an option geometry, one gets a realistic, fine-tuned model of a building.

Related to visual style, there is a chooser scenario. Inside of a chooser one can find five different possibilities, whereby four different and the last one as a synthesis of all four. Every scenario depicts one visual appearance of an agent in movement i.e. the trace he leaves. The concept behind different visual styles will be also described later.

Treatment-chain chooser enables the choice between two scenarios:

- pathway of an emergency patient
- pathway of an ambulatory patient.

A switcher is-export-on enables export of frames during the simulation. When on, the frames will be saved in a pre-defined folder.

A slider gap increases a distance between floors and makes it easier to observe and set a simulation. A button setup builds the scene, while create creates agents.

5.5. NetLogo Procedures

> **(Import building)** Schemata, personal/patient´s flow and adjacency matrix are already stored in the database. In order to get these data for simulation one must use web services. The program written in NetLogo deals with a) building's geometry and b) agent's flow.

Let us discuss first the geometry of the building. Procedure import-building creates a simulation environment. As already mentioned, there are three functionally and visually different depictions of the building. Each one of them could be a ground base for simulation depending on what do we want to show.

Figure 53 depicts building's geometry in form of colored turtles. These turtles are waypoints and openings that will be used later as a base for creating a circulation grid. Schema (one for each level), containing color-coded polygons is downloaded from web services. We have already set the limits of NetLogo´s world to x_{min}=-64 x_{ma}x=64 y_{min}=-64 and y_{max}=64. NetLogo takes the schema in form of a png image and scales it to given coordinates. As illustrated, turtles are quite large but it was necessary since 3D NetLogo had difficulties to read files in bigger resolution. Figure 54 illustrates a circulation grid. An easy way to comprehend how

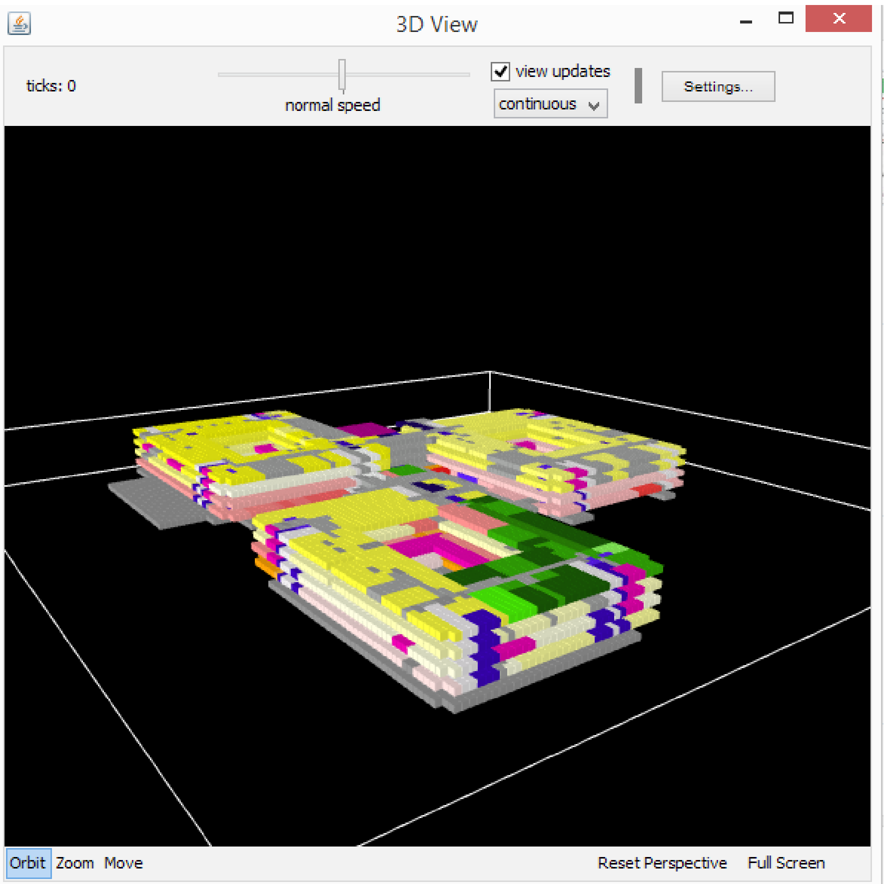


Figure 53 Geometry of the building given trough 3-dimensional cubes (NetLogo´s turtles)

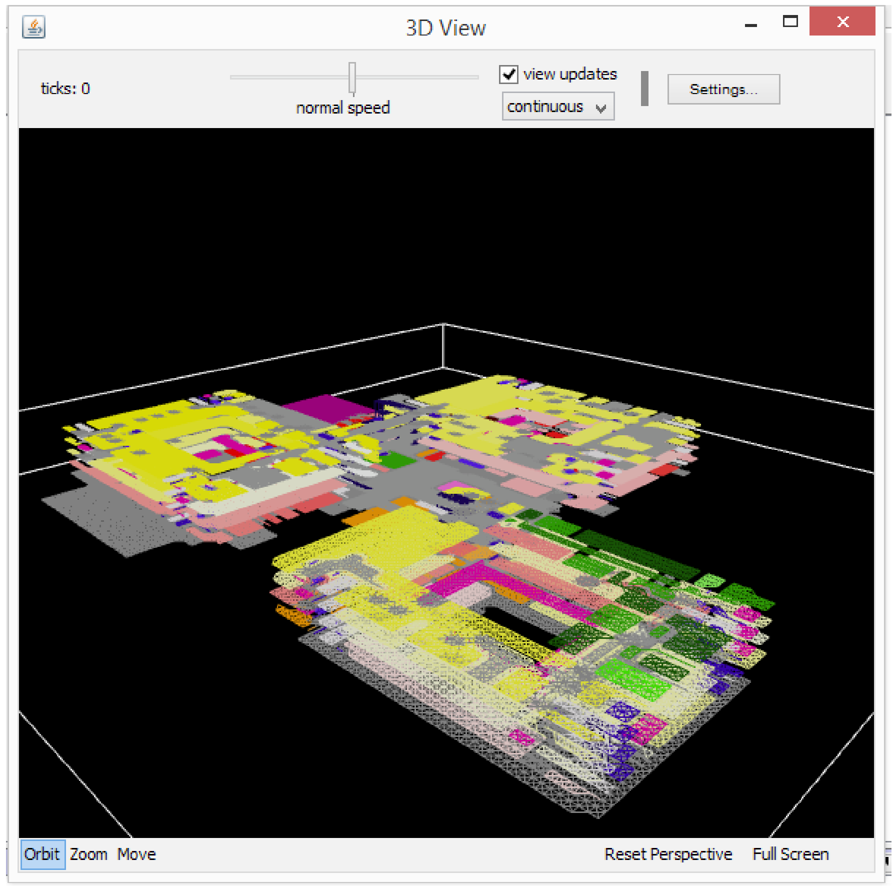


Figure 54 A grid of links that agents use for movement

agents in NetLogo move is to simple imagine a chess table. A queen in chess can move from one field toward any of eight surrounding fields. Likewise, an agent in NetLogo can move from its position toward any of eight surrounding turtles, if created.

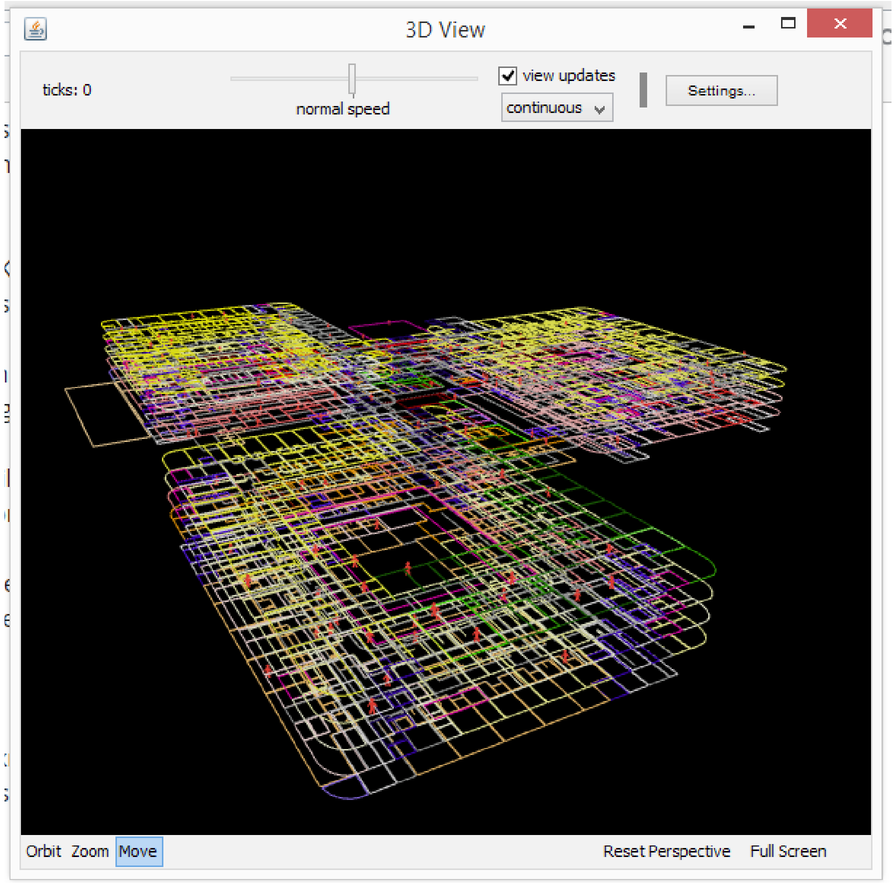


Figure 55 The geometry view depicts the geometry of buildings by redrawing the room-polygons

Assumption is only that links between turtles are created. Turtles created and linked in raster view are hidden in circulation view.

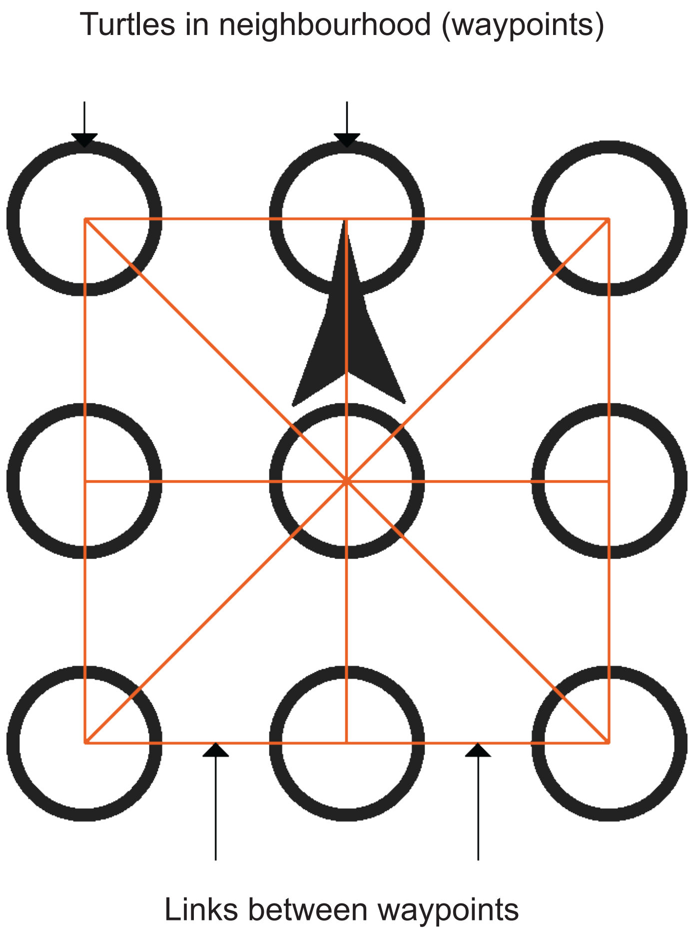


Figure 56 Agent uses links and waypoints in order to move from one position to another

Only links between them are displayed. Figure 55 illustrates a logic that enables an agent to change its position. The sketch explains only the movement on horizontal plane. One turtle has eight neighbors in xy plane, but 26 in total, if we observe the z axis as well. In order to set an agent in motion in horizontal plane, we observe only his eight neighbors. Since the building has five levels, we also need elevators, which are designed as vertical links.

Let us remember a step where we have exported drawings to the database. At that point, vectorization took place and accordingly each polygon is determined with perimeter vertices. Each vertex is defined by x and y coordinates.

When one creates geometry view, he make use of web services in order to get these coordinates. The geometry of the building is built by connecting nodes holding corresponding coordinates trough simple lines (edges). Each line receives color of matching polygon (Figure 56). Geometry view is set as default. By starting NetLogo, raster and circulation views are hidden.

5.6. Setting Agents in Motion

From a web service, we get a table of frequencies, which is stored in the database. The spreadsheet gives frequency of people flow between departments. Three values are used: 0, which stands for the lowest frequency, 0.5 the middle value and 1 which describes the high frequency. As mentioned in the Chapter 4.4.1. these numbers do not provide information on spatial relations i.e. closeness and distance between departments.

Beside mentioned spreadsheet that contains department´s names and traffic frequencies, there is another one that contains departments and their colors. This way, database provide us with all necessary information that we need for setting agents in movement. Knowing the color of each department, we write the program in such a manner that an agent chooses his starting and end point based on an information that he receives from the list.

For each frequency level, agents between departments contained in the corresponding list, will be created. For example, in case of a low frequency we will create one agent between two departments from low frequency list (Figure 57).

Frequency lists give us always the department where an agent starts his route and the department where his travel ends. Based on agent´s starting point we get the information on color of that department from the list that contains color codes for each department (Figure 45). The same we get the color of the department where the travel ends.

Figure 57 illustrates a detail of the spreadsheet with department´s names and frequencies. This table needs to be translated into several lists, which are understandable to NetLogo. From the spreadsheet, NetLogo takes the first row where the names of departments have been listed. These information are stored in one list (List A). Afterwards NetLogo takes the second raw and

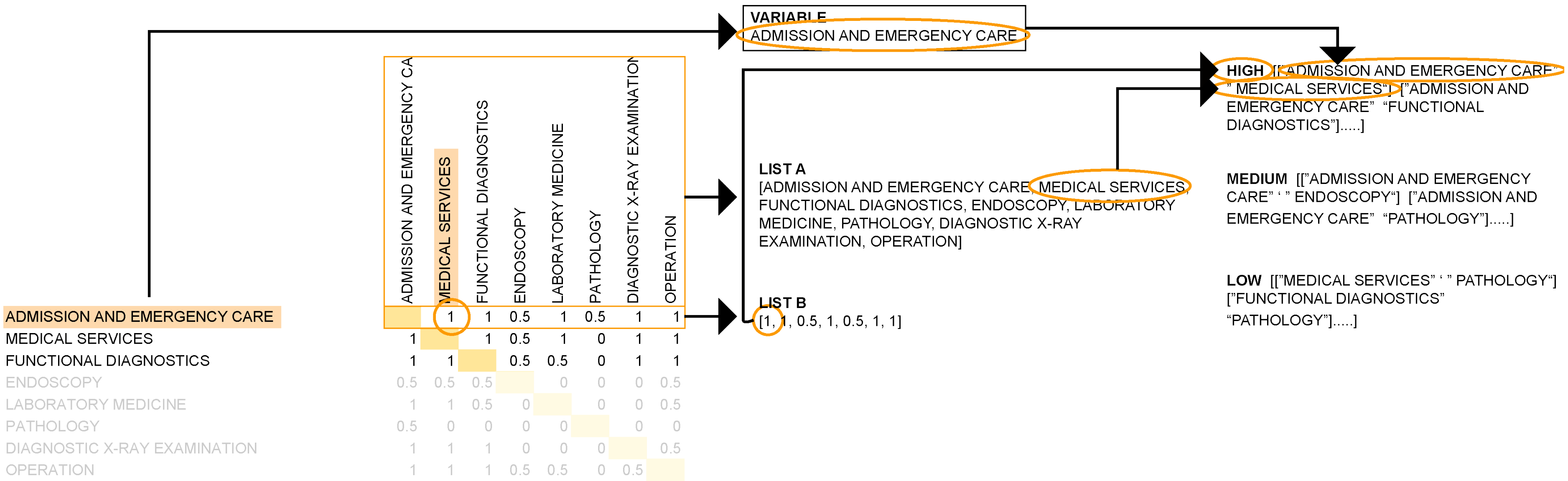


Figure 57 NetLogo converts a spreadsheet saved as .csv file to lists

puts in a second list (List B) everything but the first field. The first field contains department's name while the other fields in the row represent frequencies of people flow toward other departments. Every department from the first list have a corresponding department in the second list (frequency level). Let us assume that this number is zero. NetLogo takes the department's name from the first field of the second row and the department's name from the first row (list A). This two data are written in a sub-list C, which is a subset of a list LOW. NetLogo proceeds in a way that it reads a next field from the second row and depending on a number that it read (0, 0.5 or 1) the pairs of data will be stored as subsets in on of the lists LOW MEDIUM or HIGH. The same process flows until all rows i.e. fields are read.

In a certain department with assigned color, an agent is created. Every agent has an attribute named goals. The goals represents an array of turtles that should be crossed between start and end of his route (Figure 58).

In order to calculate a path of an agent, NetLogo picks a waypoint with the color matching to end department. With two points known and a grid of links between them, NetLogo creates agent's route using the method from NetLogo NW extension.

The same logic is valid for two other frequency lists, medium and high, with the exception that more agents will be created.

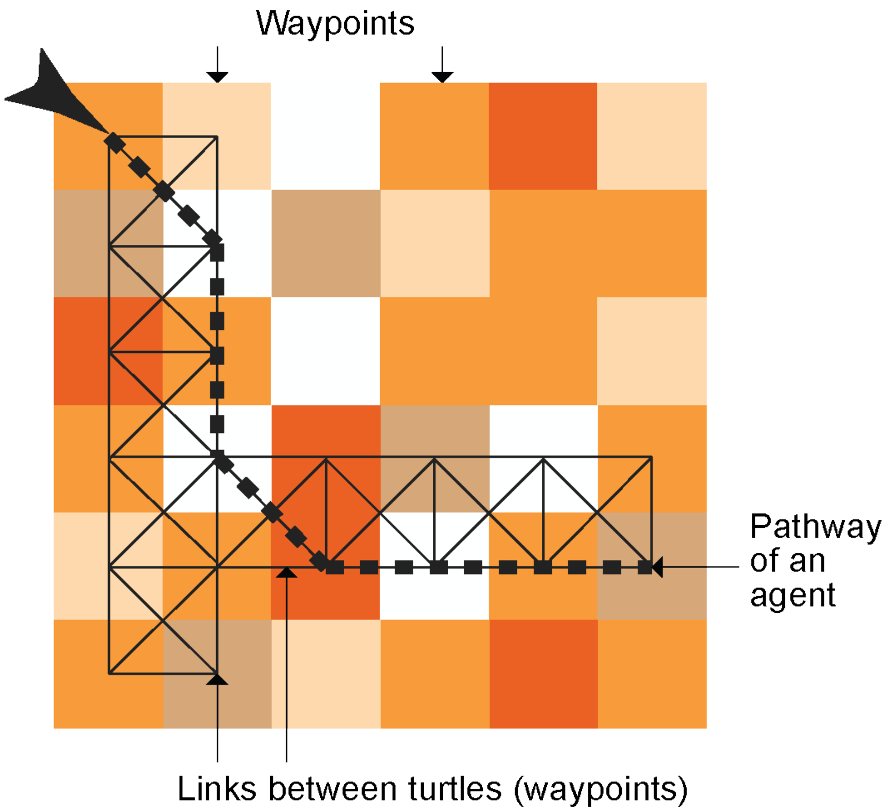


Figure 58 Path of an agent

5.7. Programming Treatment Chains

> Chapter 5.3. Treatment Chains describes treatment chain for a) ambulatory patient and b) emergency patient. Following the logic given with Figures 48-51, we translate the functional diagrams to NetLogo language.

In previous chapter, we described an agent´s transition from one point to another. We used spreadsheets, which contain department’s names, their colors and frequencies between them. With such an approach, it is possible to make connection between any two departments as long as they have different colors.

A general treatment chain given with Figures 48 and 50 can be an input for color-based programing solution. However, for a detailed pathway given with figures 49 and 51 we need other approach.

If one wants to show a patient traveling from a room to room (Figure 59), whereby rooms belong to the same department (which means that they have the same color) one needs to:

a) Have a spreadsheet with as many colors as there are rooms
b) Have an additional spreadsheet containing department’s IDs
c) To choose points of start and end as well as all the midpoints important for an agent´s movement manually.

We use the last option since it was the easiest one at the moment. In order to define a starting point, one picks manually a turtle in desired department i.e. room. Between start and end there are several transition waypoints (depicted as turtles in NetLogo). One needs to pick manually one turtle in each transition room and at the end a turtle in the last department where an agent ends his route.

By choosing all these points, NetLogo calculates a route of an agent and sets all transition turtles as well as the last turtle in the attribute goals of the agent.

It is very important to have a proper grid of links on every level of the building. The links between turtles (waypoints) are the path, imaginary road, used by agents for movement. Without links, an agent cannot travel from one waypoint to another. Vectorization produces artefacts and irregularities on schemata. This can be the result of the reduced resolution as well. For example, layer door openings, drawn in a vector software package by losing its quality, losses also a few lines that stand for doors. Without door, a room becomes an island within a schema, unapproachable from any other room or a corridor. These irregularities can be fixed manually in NetLogo, in a way that we pick two unconnected

Figure 59 Agent´s pathway from emergency entrance toward OP

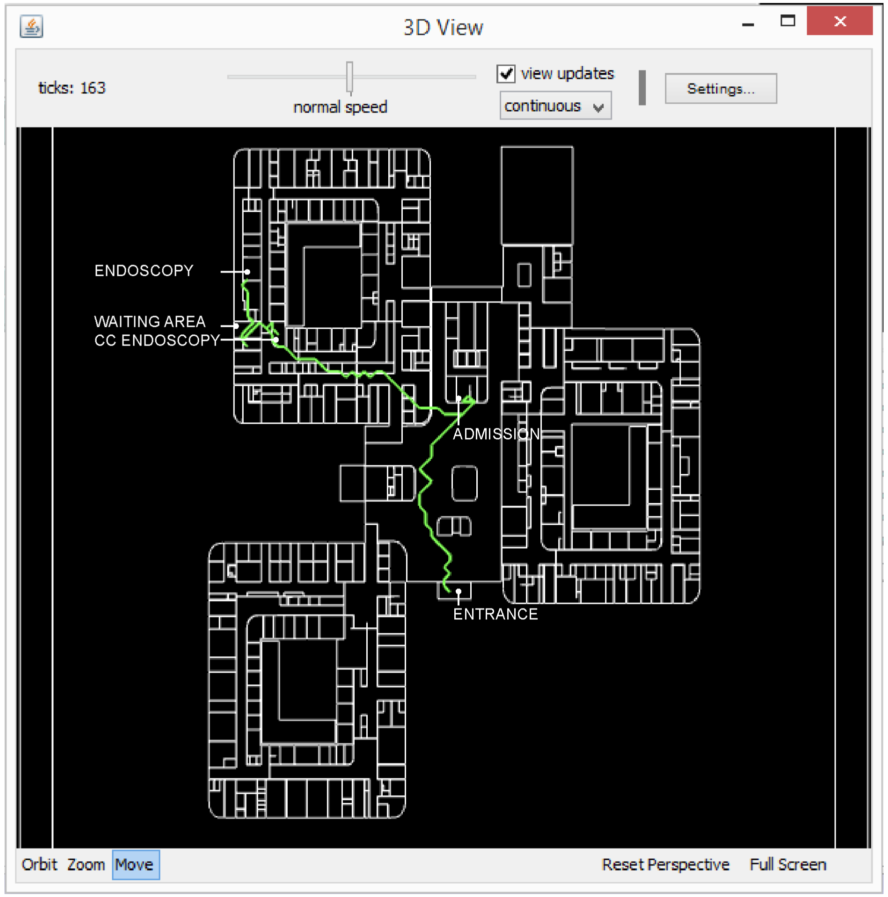


Figure 60 A pathway of an ambulatory patient

waypoints and connect them by creating a link.

Figure 60 illustrates a pathway of an ambulatory patient, admitted to the examination. If we observe his starting point i.e. entrance, it is very clear that one needs a very broad legend in order to make things work.

For example, an entrance belongs to corridors and accordingly has the same color. If one tries to set an agent in motion with starting point taken based on color from the spreadsheet (color-coded departments), an agent will take his start point anywhere as long as the color matches.

From that reason one needs to define with color every department/room that treatment chain requires or he can always do the work manually.

5.8. Visual language

> Considering a size and layout of the hospital where the simulation takes place, it was necessary to develop a unique visual language when showing a patient/personal flow. If we let only agents, depicted as persons or other shapes to move around, it would be hard to derive any conclusion from it. A lot of turtles that fly over levels does not give any information but their movement itself. It this light, four scenarios have been developed and the fifth one as synthesis of previous four.

Behid these scenarios lies an idea to provide different kind of information based on people flow only. Figure 61a depicts an agent in movement that leaves a certain trace behind. These arrows highlihts his direction. In order to avoid a bunch of arrows allover the building (since many agents are moving around) we set the trace so that it vanishes at some point. Agent’s trace in form of blue dots in various sizes illustrated on figure 61b provides visual information on crowdedness. When an agent detects another agent in neighborhood his trace, as well as the trace of the other agent becomes a larger dot. The simulation

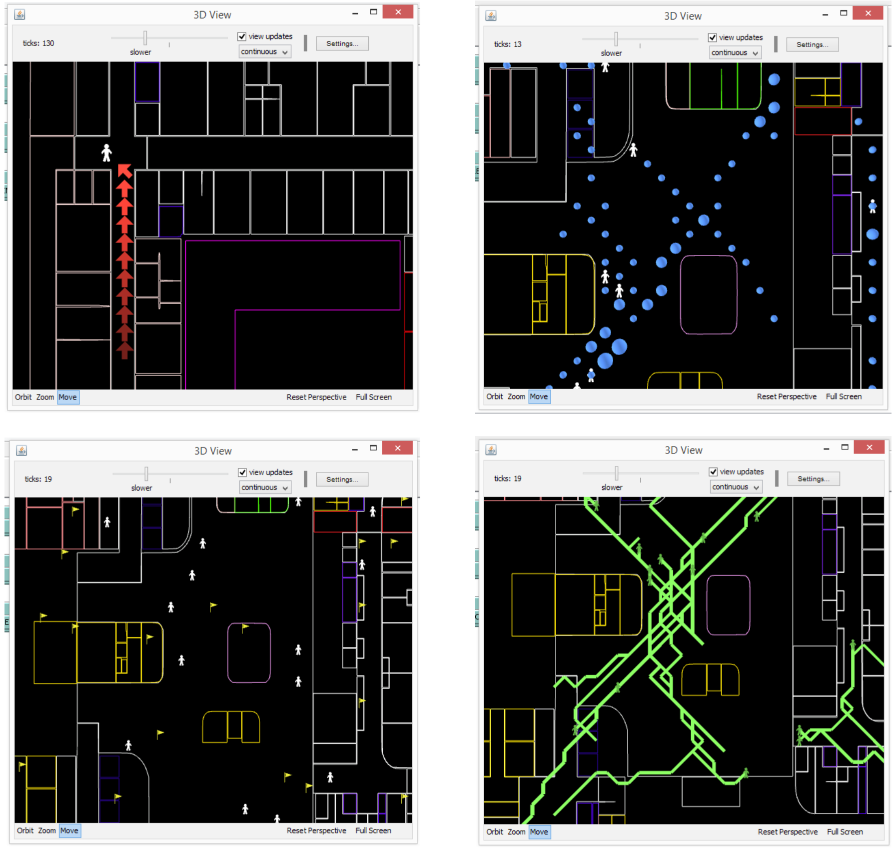


Figure 61 Simulation´s graphic styles

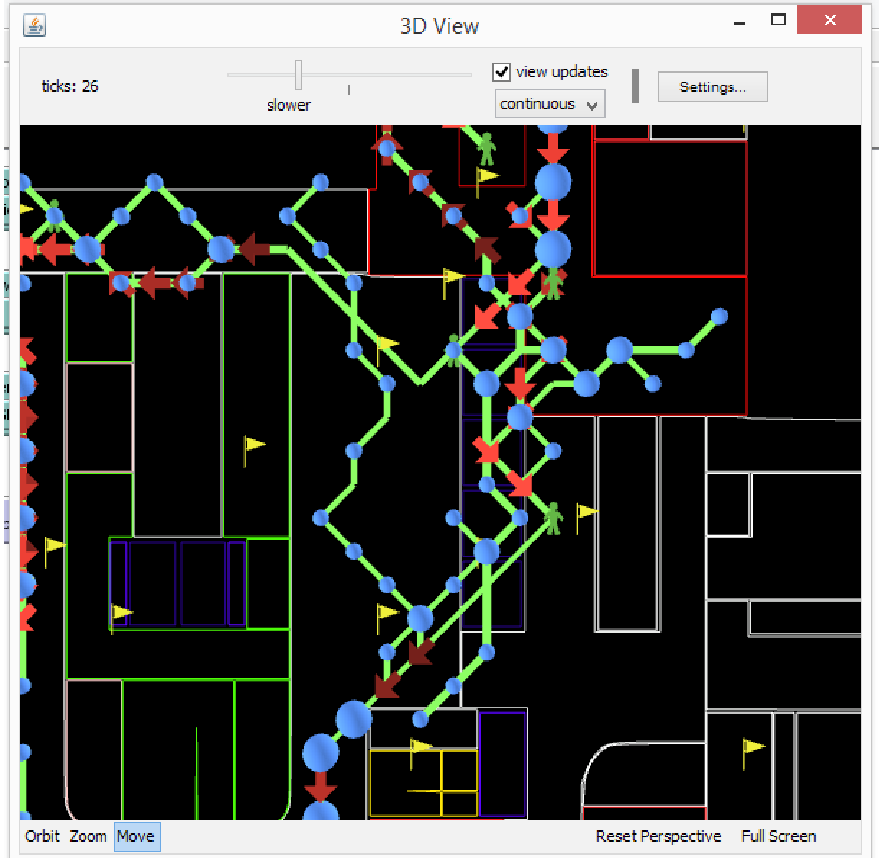


Figure 62 All four scenarios simultaneously

imitates a realistic staff and patient flow between departments. If one lets agents to move around without any features, it could be hard to establish or at least to observe where an agent begins his movement and where does he ends it. As long as we wish to depict a realistic treatment chain, it was also necessary to develop a visual style that correspond to given task. On the Figure 61c agent's start and end departments are given with yellow flags. When he initiate his motion, yellow flag is left behind at start position and when he finishes his route, yellow flag appears at the place he “dies”. Dying of agents has nothing to do with dying of patients. We made such a setup, that an agent dies i.e. vanishes from simulation windows in order to avoid a crowdedness 61d concerns with the travel lines. Striking green lines show a whole route for one agent, from start to the end of his journey.

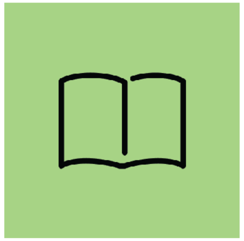
Figure 62 depicts all four scenarios simultaneously. That way, an observer collects visual impressions on crowdedness, travel lines, start-end departments and heading of agents.

1.

Introduction

2.

Computer Simulation as a Planning Tool for Architect



3.

Related Work

4.

Improved Workflow

5.

Case Study Baden

6.

Discussion

6. Final Review

In this work, we have presented the improved workflow in creating a simulation in hospital environment. We have found critical points inside of the current workflow and developed an approach how to enhance weak links. During the process, we paid attention to current State of Art.

Obtaining building floor plans i.e. their further adjustment are identified as the most problematic and the most time-consuming actions in the whole process.

We suggested the solution that could significantly reduce a total time and requires less effort at the same time.

Case Study presented in the last chapter shows an implementation of our solution in one concrete situation.

Possible applications

Simulation as a tool of architectural planning can be used in various situations.

Early phase of the project development. With new buildings, an architect can offer to programmer who develops a simulation plans in form of schemata. This is a standard phase in the project development, so an architect do not need some extra time to produce schemata for the simulation, since he already works (early phase) with schemata.

Designing a building, step by step, an architect in cooperation with a programmer, develops a simulation that helps him to control his ideas. Trough simulation, a building becomes even more comprehensible than it was using two dimensional drawings and renderings. It offers even more in comparison with animation, because it gives insight to internal mechanisms. Simulation, different from other visual representations uses also other types of data, such as number of visitors and discovers weaknesses and strength of current design. It sets relations between the building and the user and gives valuable information in return. That way architect has enough time to react on his own designs.

Renovation/adaptation of a building. The situation where our improved workflow can find a solid implementation is a renovation of an older building. There are cases when the old building plans exist and architect get these from investor. If a building is quite old, plans are usually in bad shape and even if plans are scanned, it requires still a lot of effort to make it useful i.e. to get schemata from those plans. Since the architect is already involved with the project, investor/owner can provide him access to every floor of the building (that should be renovated), he can take pictures of every fire escape plan and use it further.

Scientific purposes. Simulation can be used also as a scientific mean of analysis. In this field finds our research its fulfillment to the full extent. If an architect in an office works on a project, he usually manage to get all necessary data related to his project. If the project is about remodeling of an older building, he usually gets the plans of the existing structure. As already mentioned, the question is in which condition these plans are and how much work must one invest in further editing i.e. preparation for the simulation.

But the biggest challenge stands before researches. They are not connected (at least not necessarily) to architecture offices and they have difficulties to reach building plans. On the other hand, a simple visit to the building, taking only a camera or mobile phone, can save a significant time and energy. From this point, one takes the photographed fire escape plans and edit them as we suggested in previous chapters.

Possibilities in further development

As described in the paper A Building Database for Simulation Requiring Schemata a one further step in the development could be a database, where the hospital floor plans would be stored. Everyone could upload the building plans in form of schemata or download the data.

Using OCR can enhance the process of adjusting/editing the plans. Currently one needs to produce a spreadsheet with names of hospital departments manually. With OCR, the names of the organizational areas would be recognized and listed automatically.

OCR can be also used for recognition of single rooms. At this point if one wants to simulate a motion of agents in a hospital from one room to another, he needs all names of the rooms listed in one spreadsheet.

The logic behind tells an agent to move from one room to another using colors of the rooms, which are also listed within a spreadsheet. For such a list, one needs a lot of time. With OCR the process would be much faster.

A Building Database for Simulations Requiring Schemata

Gabriel Wurzer[1], Jelena Djordjic[2], Wolfgang E. Lorenz[1] and Vahid Poursaeed[3]

[1]TU Wien
Vienna, Austria
{*firstname.surname*}@tuwien.ac.at

[2]Moser Architects
Vienna, Austria
lela.simanic@gmail.com

[3]Iran Univ. of Science and Techn.
Tehran, Iran
poursaeed_v@arch.iust.ac.ir

ABSTRACT
Obtaining spatial representations of existing buildings for use in simulation is challenging: To begin with, getting permission to access submitted construction plans can take a long time. Then these might only be available in analog form, making it necessary to scan and vectorize them at the regulations office. The resulting representation might still not be adequate for simulation, requiring further extraction of relevant features and enrichment by additional information in order to fit the simulation domain. In our work, we have specifically targeted simulation types that work with schemata (e.g. occupancy, work and egress simulations). Our contribution lies in restructuring the aforementioned workflow so as to (1.) minimize time and effort spent on digitizing and to (2.) automatically derive schemata – sets of boundary polygons which (3.) can be further enriched by attributes. These steps are embedded into a web-based building database which allows uploads and queries per web interface as well as web services. The query interface furthermore includes (4.) the ability to download the schemata both in vector as well as raster form so that they can be used for both discrete and continuous-space approaches. Apart from acting as data provider, the database furthermore (5.) allows for spatial predicate functions which may be used for analysis of a space program.

Author Keywords
Simulation; database; spatial representation; web services.

ACM Classification Keywords
H.2.8 DATABASE APPLICATIONS / Spatial databases and GIS

1 INTRODUCTION

Schemata form the basis for a wide range of simulations that need spaces to be represented simply as boundary polygons (continuous-space simulations) or collections of cells (discrete-space simulations). Such a representation could be derived from data-rich models (i.e. Building Information Models [BIMs] for architecture; Geographical Information Systems [GIS] and Web Mapping Services [WMS] for the urban case), however, these might not be available when dealing with existing buildings for which no digital plans exist. Improving the workflow for getting a digital schema from an analog plan was thus our primary goal – technically achieved by the implementation of a web-based building database that can process images into schemata and allows for elaborate queries via web interface and web services. In more detail, we have

- looked at the current workflow for obtaining schemata from analog plans, carefully restructuring and automating individual steps so as to minimize time and effort spent (see section 'Restructuring and Automating the Digitization Process'),
- incorporated the resulting schemata into a web-based building database which lets us enrich the data further by uploading attribute tables and other media (see section 'Database Representation'),
- devised a query and data retrieval interface in the form of a web interface and RESTful webservices, providing a way for users and applications to interact with the data (see section 'Query and Data Retrieval Interface').

In order to argue for the applicability of our method, we have tested both import and retrieval using a showcase setup (see section 'Showcase'). We further provide a discussion that outlines limitations of our approach (see section 'Discussion') before concluding.

2 RELATED WORK

Many authors have already tried to infer spatial representations from image data: Koutamanis' work [2] is occupied with semantic recognition of building elements from sketches while others employ a rule-based vectorization strategy in floor plan images [3, 4]. In contrast to these approaches, we use a polygon-based vectorization algorithm [5] which has no contextual knowledge specific to architecture. Since it works on monochrome bitmaps, the user needs to separate different features by at least one pixel, which is reasonable if we assume that the interior polygon is shown and walls are simply left out.

Apart from technique used for recognition, the actual representation used for schemata might differ: For example, Tabak [6] uses a circulation graph to which he attaches spaces as single nodes. Our representation uses spaces rather than a circulation network, however we can get the schema as raster and transform the midpoint of each cell into a network node. Once connected to its neighboring nodes, the network can then be used for navigation between different spaces. Dijkstra and Timmermans [1] have used a similar approach (network of decision points), but these nodes served as intermediate goals for lattice-walking agents.

3 RESTRUCTURING AND AUTOMATING THE DIGITIZATION PROCESS

Depending on regulation, one may need a permission to access the submitted construction plans of a building. The current workflow shown left in Figure 1 thus begins with a supposedly lengthy task "obtain permission". Since we assume that we work with non-digital plans, the next step lies in a visit to the regulations office in which the plans have to be scanned or photographed and optionally stitched in a post-step. Next comes the vectorization part, which tries to identify and restore geometry. However, this process is lossy and might not be adapted to hand-drawn plans which are typical for older structures. It can thus safely be assumed that some geometry will be omitted or recognized only in part. We expect that a bit of manual work in restoring or completing features is part of the process, which also involves deleting parts of the geometry that are not relevant (task "extract relevant features"). With that, the vectorization is complete. One may still want to add attributes to certain features, e.g. room names (if not correctly recognized), functions and so on.

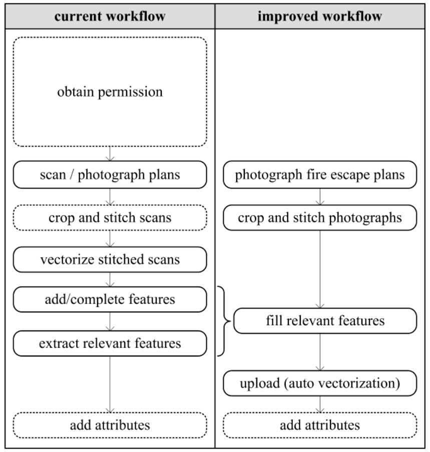


Figure 1. (left) current workflow (right) improved workflow.

When improving the current workflow, we saw our biggest challenge in the initial permission step. There are two ways in how to circumvent this, (a.) by photographing the fire escape plans without asking for permission or (b.) to scan floor plans shown in printed literature. However, the last option seems only feasible for historical architecture, since magazines and books publishing contemporary buildings tend to depict only certain areas or levels. Thus, we stick with the first case - to photograph fire escape plans (see right in Figure 1) since we aim at recent architecture. Because our photographs will likely contain areas and not a whole floor, we must stitch them together in a post step.

The biggest difference between the current and improved workflow comes in the next step: Rather than vectorizing and completing features in a post-step, we fill each extracted space (a simple flood-fill operation in a drawing package) with a known color (or a set of colors, if multiple space are to be marked as belonging to one common area). Details that are not relevant (e.g. doors, equipment) may need to be cleared beforehand since the schema is only concerned with boundaries. A further way in which superfluous details can be eliminated is to clear all pixels that are not in the range of colors used to signify spaces.

The color-coded images are now uploaded to the database which performs an automatic vectorization. In that process, it assigns an id and color to each resulting feature. The vectorization we use guarantees that all features are found if there is at least one pixel between separate elements. The result is registered as vector layer of a certain name (e.g. "schema") at a certain level (e.g. 1) of a certain building. We use naming conventions on the image file in order to extract this information.

In a post-step, one may assign attributes to individual spaces or areas by specifying a mapping of the form (color, ...*attributes*...) or (feature id, ...*attributes*...) interactively or by upload of a spreadsheet.

4 DATABASE REPRESENTATION

The database acts as a repository that stores uploaded schemata per building and level. It furthermore offers the possibility to upload other media (e.g. images, spreadsheets, documentation) which are registered on a per-building basis. An graphical outline is given in Figure 2.

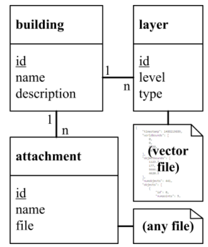
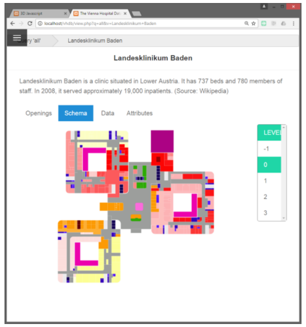
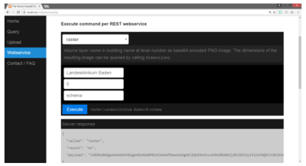


Figure 2. Database representation and linked files.

Technically we use the sqlite database to store all metadata concerning buildings, layers and attachments. The actual vector data is represented as files containing Javascript Object Notation (JSON), attachments are written out as files in their original formats.



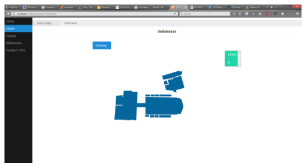
(a)



(b)

Figure 3. (a) Web-based query interface with 2D content viewer. (b) Web service call with results shown online.

(a)



(b)

Figure 4. (a) Touched-up fire escape plans and (b) 2D viewer displaying the ground floor of our institute.

5 QUERY AND DATA RETRIEVAL INTERFACE

Until now the changed workflow proposed by the authors does little more than to vectorize image data with some added constraints (1 pixel separation between spaces) such that these become distinguishable. The true power of the building database lies in leveraging the information for producing extracts in both raster and vector form, offer spatial predicates based on the imported data and to be interfaced in multiple forms, either

- through a web interface which allows end-users to query and view content in 2D (see Figure 3a) or
- a web service which enables applications to interact with the database (see Figure 3b for a web-based service call and response).

The interaction lies in mapping/transforming geometries to the client coordinate system (by scaling the whole geometry or extracting a portion, given by a window) in both vector and raster forms (by rendering geometries to a raster map), and to map colors and indices to attributes (through attachments in spreadsheet form listing [color, *...attributes...*] or [feature id, *...attributes...*]). Another area of queries lies in spatial predicates which are concerned with topological queries (e.g. neighboring spaces), realized internally through the use of an adapted form of the Dimensionally Extended nine-Intersection Model (DE-9IM) which is common in GIS systems.

The query interface does not carry a state, which is assumed to be supplied by the client. In more technical detail, our web service expects to be called in RESTful form, through

{function name}/parameter1/parameter2/.../parameterN

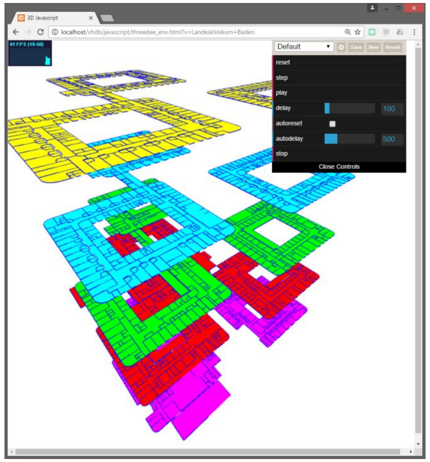
which is received by a PHP stack and mapped to a range of service routines. The results of each call are given in JSON, giving a wide range of web-based clients the opportunity to interact with our database.

6 SHOWCASE

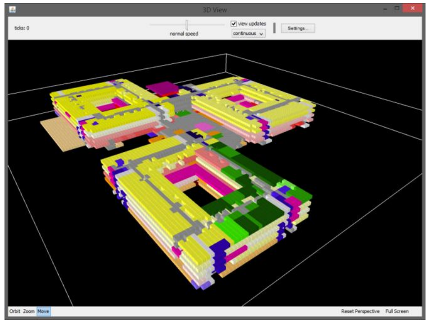
In order to argue for the applicability of our method, we have gone through all of the proposed steps and got a benefit in terms of time spent digitizing and effort involved in getting the gathered data ready for simulation. Our test was split into two phases, (1.) gathering input using fire escape plans, which were digitized automatically and (2.) writing client programs that would access data stored in the database.

For the first task, we scanned the fire escape plan of our own institute, which took 28 minutes for outlining a single floor. The touched-up data is shown in Figure 4a and 4b and is discussed in due course (see 'Discussion').

For the second task, we imported a building from existing floor plans. A data query in vector and raster form was instantaneous, in the order of milliseconds. It enabled programs (see Javascript and Netlogo 3D in Figures 5a and 5b) to utilize a rich scale of options to choose from when mapping into an own coordinate space and, at the same time, use the set of spatial predicates available via the data interface of the database.



(a)



(b)

Figure 5. (a) Javascript [three.js] and (b) Netlogo 3D viewer accessing the database via web servives.

7 DISCUSSION

Apart from all technical assumptions, it is clear that we presume that we can enter all levels of a building for photographing fire exit plans. This optimistic strategy is driven by the fact that a simulation study is typically conducted in a project context where the client actively supports the retrieval of plans.

The second discussion point lies in the contribution of the method. It is clear that some of the mentioned concepts could be achieved by redrawing features in a vector program, however, the proposed service architecture offers additional functions that make the proposed system valuable as content provider, freeing simulation of having to implement certain functionalities on their own. With that, we are aiming at a service architecture enabling client programs to scrap some of their code when connecting to databases, which is a good outlook for the development of the field.

8 CONCLUSION

We have presented an improved workflow for obtaining digital schemata from analog images. The proposed process is integrated into a building database that offers additional functionalities for querying, transforming and annotating that data and is accessible via a web interface and web services. The applicability of our method was showcased by conducting all steps from data acquisition to usage of the data in a simulation.

REFERENCES

1. Dijkstra, J., Timmermans, H. Towards a multi-agent model for visualizing simulated user behavior to support the assessment of design performance, *Automation in Construction 11*, 2 (2002), 135–145
2. Koutamanis, A. Recognition of Building Elements in Free-Hand Sketches. *Proc. 26th eCAADe*, 2008, 419-426.
3. Lu, T., Yang, H., Yang, R., Cai, S. Automatic analysis and integration of architectural drawings, *Int. Journal of Doc Analysis and Recognition 9*, 1 (2007), 31-47.
4. Macé, S., Valveny, E., Locteau, H., Tabbone, S. A System to Detect Rooms in Architectural Floor Plan Images, *9th IAPR Int. Workshop on Document Analysis Systems*, 2010, 167-174.
5. Selinger, P. Potrace: a polygon-based tracing algorithm (2003). http://potrace.sourceforge.net/potrace.pdf. As of 1st March 2017.
6. Tabak, V. *User Simulation of Space Utilisation*, PhD Thesis, Technical University Eindhoven, 2008.

8. List of Figures

List of Figures

9. Glossary

CAD	Computer-Aided Design
BIM	Building Information Modelling
GIS	Geographical Information Models
HVAC	Heating, ventilation and air conditioning
GFA	Gross Floor Area
NFA	Net Internal Area
DES	Discrete-event
ABS	Agent-based
SD	System dynamics
OSM	OpenStreetMap
VGI	Volunteered geographic information
EDM	Electronic Document Management
HIS	Hospital Information System
ERP	Enterprise Resource Planning
GUI	Graphical User Interface
DB	Database
PDF	Portable Document Format

10. References

1. George W.M.Harrison, Vayos Liapis, 2013: “Performance in Greek and Roman Theatre”, Brill, Boston.

2. August Mau, 2014: “Pompeii: Its Life and Art”, Literary Licensing, Whitefish.

3. Britannica (accessed 16.07.2017): “Linear perspective”.

4. Arthur Zajonc, 1993: “Catching the Light: The Entwined History of Light and Mind”, Oxford University Press, Oxford.

5. Vitruvius (translation by Morris Hicky Morgan), 1914: “The ten books on architecture”, Harvard University Press, Cambridge.

6. Britannica (accessed 16.07.2017): “Louis Sullivan”.

7. Elias W. Hans, Mark van Houdenhoven, P.J.H. Hulshof, 2011: “A framework for health care planning and control”, Memorandum 1938, Department of Applied Mathematics, University of Twente, ISSN 1874-4850, Enschiede.

8. William B. Nordgren, 2002: “Flexsim simulation environment”, In Proceedings of the 34th conference on Winter Simulation: exploring new frontiers (WSC ‘02), Winter Simulation Conference.

9. C. Dennis Pegden and David T. Sturrock, 2009: “Introduction to Simio”, In Proceedings of the 2010 Winter Simulation Conference, ISBN 978-1-4244-9865-9, Baltimore.

10. James L. Peterson, 1981: “Petri Net Theory and the Modeling of Systems”, Prentice-Hall, ISBN 0-13-661983-5, New York.

11. Uri Wilensky, 1999: NetLogo, http://ccl.northwestern.edu/netlogo/, Center for Connected Learning and Computer-Based Modeling, Northwestern University. Evanston, IL.

12. W Lorenz, M. Bicher, G. Wurzer, 2015: “Adjacency in hospital planning - Using Newton´s differential equation”; Proceedings of MathMod 2015, IFAC Online Conference Paper Archive.

13. E. Neufert and P. Neufert, 2012: ”Architect’s Data (4th Edition)”, Wiley-Blackwell, New Jersey.

14. L. P. de las Heras, D. Fernández, E. Valveny, J. Lladós and G. Sánchez, 2013 “Unsupervised Wall Detector in Architectural Floor Plans,” 12th International Conference on Document Analysis and Recognition, Washington.

15. Sébastien Macé, Hervé Locteau, Ernest Valveny, and Salvatore Tabbone, 2010: ”A system to detect rooms in architectural floor plan images”, In Proceedings of the 9th IAPR International Workshop on Document Analysis Systems (DAS ‘10), New York.

16. M.M.Chang, S.Or, K. Wong, Y. Yu, 2005: “Highly Automatic Approach to Architectural Floorplan Image Understanding & Model Generation”, published in journal Pattern Recognition.

17. G. Wurzer, W.E.Lorenz, M.Pferzinger, 2012: “Pre-Tender Hospital Simulation Using Naive Diagrams As Models”, in Proceedings of the International Workshop on Innovative Simulation for Health Care, ISBN 978-88-97999-13-3.

18. E.T. White, 1986: “Space Adjacency Analysis: Diagrammaing Information for Architectural Design”, Architectural Media.

19. G. Wurzer, W.E.Lorenz, 2013: “From Quantities to Qualities in Early-Stage Hospital Simulation”, in Proceedings of IWISH 2013.

20. V.Tabak, 2008: “User Simulation of Space Utilisation”, PhD Thesis, TU Eindhoven.

21. V.Tabak, B. de Vries, J.Dijkstra, 2008: “Validating an office simulation model using RFID technology”, in Proceedings of the 9th International Conference on Design & Decision Support Systems in Architecture and Urban Planning, Eindhoven.

22. R. Goldstein, A. Tessier and A. Khan, 2010: “Schedule-Calibrated Occupant Behavior Simulation”, in Proceedings of SimAUD 2010.

23. D. Simeone, Y.E. Kalay, D. Schaumann and S.W.Hong, 2012: “ An Event-Based Model to Simulate Human Behaviour in Built Environments”, Proceedings of eCAADe 2012 (Volume 1).

24. A. Koutamanis and V. Mitossi, 1992: “Architectural computer vision: automated recognition of architectural drawings”, in Proceedings of the 11th IAPR International Conference on Pattern Recognition, ISBN 0-8186-2910-X.

25. Tong Lu, Chiew-Lan Tai, Feng Su, Shijie Cai, 2005: “A new recognition model for electronic architectural drawings”, Computer-Aided Design Volume 37, Issue 10.

26. Yasuhiro Aoki, Aki Shio, Hiroyuki Arai and Kazumi Odaka, 1996: “A Prototype System for Interpreting Hand-Sketched Floor Plans”, in Proceedings of the 13th International Conference on Pattern Recognition, ISBN 0-8186-7282-X.

27. Sheraz Ahmed, Marcus Liwicki, Markus Weber, Andreas Dengel, 2012: “Automatic Room Detection and Room Labeling from Architectural Floor Plans”, 10th IAPR International Workshop on Document Analysis Systems, ISBN 978-0-7695-4661-2.

28. Sheraz Ahmed, Marcus Liwicki, Markus Weber, Andreas Dengel, 2011: “Improved Automatic Analysis of Architectural Floor Plans”, International Conference on Document Analysis and Recognition, ISBN 978-07695-4520-2.

29. Ernest Valveny, Enric Mart, 1999: “Application of Deformable Template Matching to Symbol Recognition in Hand-written Architectural Drawings”, in Proceedings of the 5th International Conference on Document Analysis and recognition, ISBN 0-7695-0318-7.

30. Moustafa Alzantot, Moustafa Youssef, 2013: “Demonstrating CrowdInside: A System for the Automatic Construction of Indoor Floor-plans”, IEEE International Conference on Pervasive Computing and Communications Workshops, ISBN 978-1-4673-5077-8.

31. Dr Christian G Huber, Andres R Schneeberger, Eva Kowalinski, Daniela Fröhlich, Stefanie von Felten, Marc Walter, Martin Zinkler, Prof Karl Beine, Prof Andreas Heinz, Prof Stefan Borgwardt, Prof Undine E Lang, 2016: “Suicide risk and absconding in psychiatric hospitals with and without open door policies: a 15 year, observational study”, The Lancet Psychiatry Volume 3 No. 9.

32. Eva-Marie Nosal, 2008: “Flood-fill algorithms used for passive acoustic detection and tracking”, New Trends for Environmental Monitoring Using Passive Systems, ISBN 978-1-4244-2815-1.

33. Gabriel Wurzer, Wolfgang E. Lorenz, 2016: “A Case Study of Social Network Analysis in Adjacency Graphs”, in Proceeding of the 34th International Conference on Education and research in Computer Aided Architectural Design in Europe, At Oulu, Volume: 2, ISBN: 9789491207112.

34. Wolfgang E. Lorenz, Martin Bicher, Gabriel X. Wurzer, 2015: “Adjacency in hospital planning”, IFAC-PapersOnLine, ISSN: 1474-667.

35. Liang Chen, 2012: “Agent-based modeling in urban and architectural research: A brief literature review, Frontiers of Architectural Research Volume 1, Issue 2.

36. Robert Wischer, Hille Rau, Karl Krämer Verlag, 1988: “Das Friesen-Konzept für das Krankenhaus und für das Gesundheitswesen von morgen”, ISBN: 3782840011.

37. D.Dunn, 1997: “Responsibilities of the preoperative holding area nurse”, AORN Journal Volume 66, Issue 5.

38. Peter Lohfert, 2005: “Methodik der Krankenhausplanung”, Lohfert & Lohfert, Hamburg.