To my mother,
顏綉鸞
Xiu-Luan Yan
DIPLOMARBEIT

Permeation

ausgeführt zum Zwecke der Erlangung des akademischen Grades eines Master-Ingeneurs unter der Leitung von

Prof. William Alsop
E253/4
Institut für Architektur und Entwerfen

eingereicht an der Technischen Universität Wien
Fakultät für Architektur und Raumplanung

Ching-Hua Chen
0727515
Favoritenstr. 62/20, 1040 Wien

Wien, am
Wittgenstein’s philosophy has been the guide of my thoughts in architectural design. Beginning from many years ago, thanks to Prof. Phoenix Ho, without whose inspiration I would never start to read “Tractatus logico-philosophicus” and to understand architecture through its point of view. The way was long and rough, but it comes around. Architecture to me is an accumulation of one’s experiences, either in space or in life, and the only way to form one’s own theory is through many discussions and debates between different perspectives. Without those arguing moments on topic “architectural design” with my college friends, among those Shu-Yu Wu and Michael Lin, my philosophical understanding in architecture would never start to form.

Starting in 2009, my architectural study extended into the field of algorithm. Computation didn’t come to me as a tool, but a possible media between theory and practice. Without those discussions and burning the mid-night oil with a true friend, Emanuel Bührle, I would not be able to even begin to understand how to program, and not even mention to come to find a relation between architecture, philosophy and computation together.

Many thanks to the late Prof. Kari Jormakka, who led me breaking through the difficult time of forming the theory. A great gratitude to Prof. William Alsop, who has always motivated me to hold on to reach what I wanted, even when the progress was hard and heading to an unforeseeable result.

Gratitudes to my families, especially to my father, I’m profoundly indebted to his unrepayable supports; thanks to all my friends in Zeichensäle Arsenal, among them, Olga Muskala, Natalia Diadus, Tamara Egger, Thomas Marian, Johannes Langer.

An immense gratitude to Johanna Linsberger. I’m deeply in owe to her unconditional supports and her company all the way through the tough final completion of my thesis.

Vienna, March 16th, 2014
DEFINE

class Architecture:
    def __init__(self, form = None, function = None):
        self.form = form
        self.function = function
    def _design(self, star, end):
        result = []
        for i in range (unlimited):
            if start == self.form:
                end = self.function
            elif end == self.function:
                start = self.form
            else:
                result.append(sqrt(form**2 + function**2))
        return result

EXECUTE

>>> building = Architecture()

RESULT

>>> result = building._design()
Abstract

In analytical way, it explains backwards,
how he came to these conclusions, by his ‘picture theory of language’.

“Wittgenstein suggested that a meaningful proposition pictured a state of affairs or atomic fact. Wittgenstein compared the concept of logical pictures with spatial pictures.”

As design deals with thoughts, and architecture involves in space, geometry and visual pictures, through Wittgenstein’s philosophy, we might be able to interpret the form/function principle with a fresher mind.

In the extension of his analytical philosophy, the composition of architecture makes a hierarchical structure. Built by systems, usually, this structure is described in objects: furniture pieces rest within rooms, which inhabit in buildings, and beyond architecture, the buildings compose districts, then cities, countries, continents, earth ... until the universe. However, described in this way, there is one thing missing - humanity. “Die Welt zerfällt in Tatsachen”: it is the facts we do that are essential. Then this structure should be slightly modified to - for example: a chair to sit rests within a room for receiving guests, which inhabits in a house to live, and many houses to live compose a district for dwelling... and so on. Beside the shown multi-level system hierarchy, two questions emerge: how the objects are and how they function to fit.

Computation has been widely utilised in many fields of architecture. From digitalising the drawings, simulating and animating models, evaluating cost and time, until recently parameterising environmental data values, construction methods or material planning within the design process. The relationship between programming and design has been gradually changing from “with the tools for the goal” to “beyond the tools and the goal”.

As algorithm is written in languages, and geometry is the basic of forms, structuralised programming within modelling reflects to applying an analytical method in philosophy as in geometry. Since we could see that the multi-level system in architecture, which can correspond to a hierarchy in geometrical forms, a new thinking of ‘from function to form’ could be found in the combination of analysed human activities and algorithmic geometries.

Tetrahedra are a basic, the simplest and a structurally stable volum. Composed by four with each other adjacent triangles - the simplest surface geometry - a representative of one tetrahedron can be reduced to its four vertices.

Truss systems consist of triangular units, on each of whose members the tensile and compressive forces are balanced, when calculated the three members altogether. If considering the unit three-dimensionally, tetrahedron truss forms the simplest space truss system, which consists of six member meeting at joints.

A multi-level system in construction, so as in its algorithmic structure, is composed of a single-triangular-panel level, a single-tetrahedron-level and a whole tetrahedron-truss-form level - a truss system composed by numerous tetrahedron units, which gradually differentiate in their size abiding their spatial orders.

Through a passage between the original and its replica of the Portal IV facade, there flows the history. Time is essential, a main factor in our lives which secures these facts indeed have happened. Time flows, carries the stories away, but left the memory of two buildings behind, Berliner Schloss and Palast der Republik, along with the new one to come - Humboldt Forum.

Three buildings on one same site in different times, left but two portals simultaneously existing - starting in a short future. Passing through this corridor, we find many stories overlapped in this one space. In the time of Berliner Schloss, the portal led directly into the I. Innenhof (1st. Courtyard), passing by the connecting building of the two more than 100 metres long facades. Where the connecting building was, was approximately the location of the memorable facade of the Palast der Republik, an image we have learnt with its shiny golden window glasses, reflecting in a picture taken in a sunny afternoon.

To be called as “Humbold Forum”, the new - now, in 2013, still in construction - building has the same old three facades - of North, West and South - and stands on the same location. Two of its five replicated portals, the II. and the IV., build a historical channel flowing to the very original Portal IV.

Whatever has passed, must have left something behind. What will we see, when we pass this corridor and leave it behind?
Standing on the two ends of the scale of design decisions, as an ultimate-stable principle, form and function have been the main debate through decades in the architectural design history.

Which comes first? Should forms still follow functions? If so, do those advanced form languages misspell in their own grammar? If not, wherefrom should a form be derived?

Or as some would say, functions are not important for architecture. However, we must not confuse the functionality of an architecture with the names of functions. A name can be endowed to an object - in this case, a building - only when and if it functions as how the name indicates. Thus, a building loses its being as an architecture, once it cannot satisfy its “societal function as the innovative framing of social interaction”.


Thoughts of this section are inspired from the following texts, books:

“Tractatus logico-philosophicus”
- by Ludwig Wittgenstein, 1921;
- English translation by C. K. Ogden

“The autopoiesis of architecture” Vol. II
- by Patrik Schumacher, Chap 6 about functions

Keywords:
Form and Function, Wittgenstein,
EXPRESSIONS

along with signs and their configuration are the elements in a proposition. But how is it related to a thought? 4 Der Gedanke ist der sinnvolle Satz. Spatially, if out thoughts obey the logic in space, i.e. of geometry, then what we say, i.e. our propositions, wherefrom our expressions, have sense.

3.31 Der Satz selbst ist ein Ausdruck. Philosophically, the German word “Satz” is translated as “proposition” in C. K. Ogden’s version of Tractatus logico-philosophicus. A proposition is “a statement or assertion that expresses a judgement or opinion”. Our thoughts, judgments or opinions are described by expressions in propositions. “Satz”, on the other hand, means also “sentence”, where the signs - if simply explained, as the meanings we sense in words - are ordered with a configuration to form a sense.

However, signs are not equal to words. We cannot just mirror from one to another. Signs are those what we want to say behind what we write. What we write are symbols, which share with each other what is expressed in the proposition. 3.5 Das ange-wandte, gedachte Satzzeichen ist der Gedan-ke. We understand each other’s thoughts through the languages by the sentences we make, and we comprehend what the sen-

tences express by the propositional signs (Satzzeichen).

Making thoughts into sentences in languages is one way of expression. In design process, expressions of the propositions, i.e. the sentences, show our intentions to create a project. Intentions make the sentences not just a combination of words, but into propositions to sense. We speak of those elements that matter in the project, and organise them into subjects, objects or predications in their corresponding positions, according to the language syntax, within the descriptions of our thoughts.

Considering a spatial arrangement, examplarily we would simply think: “An altar stands in front of the audience.” One sentence is one thought. One spatial proposition make one place. “3.4 Der Satz bestimmt einen Ort im logischen Raum.” As long as the proposition follows the logic of space. This sentence carries a thought implying a main function of church: preching. However, the question would be: how the sentence makes sense, without mentioning preching? An altar relates us to religions, and the audience to the religious, but without being articulated by a spatial coor-
dination “stands in front of”, they are just two opposite sides of groups: the givers and the receivers, and the space does not yet exist.

Yet surely, if no word of preching were actually mentioned, there can only be implications that this activity could happen here in this spatial arrangement. It is however the tasks for architects, to provide the possibility by coordinating in space. The under-

standing of an implication, is important.

Certainly, a word like altar indicates in many ways. We receive the signs when we read the word. “3.32 Das Zeichen ist das sinnlich Wahrnehmbare am Symbol.” It indicates that there is someone standing at this spot on a higher level speaking and which is something religious. But how do we understand its manifold meanings in just one word?

Words are symbols. When we see the signs, we perceive what they signify. “3.326 Um das Symbol am Zeichent wahrnehmen zu müssen, muss man auf den sinnvollen Gebrauch achten.” And certainly, to further use the symbols for our expressions, as this sentence here, means we have already defined them by our usages. Altar is a word signif-
cing certain meaning, defined as “an elevated place or structure, as a mound or platform”, more important, “at which religious rites are performed [...]”. One certain meaning dis-

Distinguishes it from just being a structure, a mound or a platform, is ‘religious’, which focuses where this proposition could take place, such as a temple or a church. The counterpart of altar is the ‘audience’. Without further confusion of its broader defini-
tions, the audience here along with altar, targets on meaning the religious listeners or viewers.

Having only ‘altar’ and ‘audience’ does not make a sentence. Without the pred-

ication ‘stands in front of’, they have nothing to do with each other. “3.327 Das Zeichen, das Zeichen bestimmt erst mit seiner logisch-syntak-
tischen Verwendung zusammen eine logische Form.” To follow the syntax, we express cor-

rectly our thoughts, it makes a configuration of the symbols we use in the proposition. Clearly, we can also say “The audience stands in front of an altar.” It is actually syntactically correct, and it does have sense. However, it might convey a different sense as we design for, since the audience might then turn their back to the altar, and be standing, contrary to be seated as what we usually might have in mind, where the altar is the one situated higher.

“3.411 Der geometrische und der logische Ort stimmen darin überein, dass beide die Möglichkeit einer Existenz sind.” If a proposition is true or false, we have to check with reality, i.e., in the spatial sense, the cor-

modation of geometries. “3.41 Das Satzzei-

chen und die logischen Koordinaten: Das ist der logische Ort.” The logical space contains all the possibilities of the propositions which have sense, as these two sentences both have sense, but it depends on if the coordination could make this space function correctly, that we decide which design to realise with the corresponding object in reality. “3.2 Im Satz kann der Gedanke so ausgedrückt sein, dass den Gegenständen des Gedankens Ele-

mente des Satzzeichens entsprechen.” How-

ever, how do we connect which object (Ge-
genstand) corresponding to which elements we have in mind?

6 “The thought is the significant proposition.”
7 “A proposition itself is an expression.”
8 http://www.oxforddictionaries.com/us/definition/ american_english/proposition
9 “The applied, thought, propositional sign, is the thought”
10 “The proposition determines a place in logical space: […]”
11 “The sign is the part of the symbol perceptible by the senses.”
12 “In order to recognize the symbol in the sign we must consider the significant use.”
13 http://dictionary.reference.com/browse/altar?r=1
14 “The sign determines a logical form only together

with its logical syntactic application.”
15 “The geometrical and the logical place agree in that each is the possibility of an existence.”
16 “The propositional sign and the logical co-ordinates: that is the logical place.”
17 “In propositions thoughts can be so expressed that to the objects of the thoughts correspond the elements of the propositional sign.”
Picturing

objects in mind is making thoughts of their coordination with each other as the fact how they function. "3. Das logische Bild der Tatsachen ist der Gedanke." 12

Take the thought of "An altar stands in front of the audience.", for example. This specific object, an altar, in this proposition, we are actually picturing an object with this specific definition, 'an elevated place or structure [...] at which religious rites are performed [...]." This religious performance distinguish this object - which the word 'altar' refer to - from being just a structure or platform. If the picture in our mind has an 'altar' object, then we have already formed a religious situation intentionally. Thus, the picturing process is not related to the object form but to the idea of, in this case, its 'being an altar'. "2.13 Den Gegenständen entsprechen im Bild die Elemente des Bildes.« This idea of 'being an altar' is called an element of the picture, as its counterpart 'being audience' in this picture as well. So, if we are picturing of this certain 'altar' or 'audience', we are not thinking of the objects themself but the a situation where their functions are also involved - either for a religious performance or being listening or viewing.

"2.14 Das Bild besteht darin, dass sich seine Elemente in bestimmter Art und Weise zu einander verhalten." 20 These elements - the altar and the audience from this example - are combined in this way that one 'stands in front of' another, which composes this one picture, which is a fact as well. "2.141 Das Bild ist eine Tatsache." 21

What leaves us for those objects - the elevated level platforms where the priest preaches, commonly called altar - or - those people listening or the objects where those listeners sit on, where 'audience' refers to? "2.0232 Beiläufig gesprochen: Die Gegenstände sind farblös." 22 These objects are those which carry the certain forms - as in space - but without religious performances, listening or viewing - as Wittgenstein called them: 'colourless'.

In the conceived pictures, there is a form. "2.023 Diese feste Form besteht eben aus den Gegenständen." 23 There are the forms of objects, as seen as the elevated level or as the seats to be sitted on, but there is also the form as the whole picture appears. "2.151 Die Form der Abbildung ist die Möglichkeit, dass sich die Dinge so zu einander verhalten, wie die Elemente des Bildes." 24 The form here does not just mean the geometrical appearance of an object, such as an one and a half metre elevated platform or a fifty centimetre high wooden board to sit on, but the form of an object as how its being as it is, '2.0251 Raum, Zeit und Farbe (Färbigkeit) sind Formen der Gegenstände", and the form of the coordination of the objects as well.

Every picture we make in mind, makes a possible situation. "2.203 Das Bild enthält die Möglichkeit der Sachlage, die es darstellt." 25 As we architects design, we search for possible solutions in space, for which we illustrate them with our imaginations in forms. What actually compose these possibilities are not those form-objects but those pictorial elements - e.g. the idea of an altar or the audience - and along with their corresponding coordination - in this example, it is the 'stands in front of'.

The two essential constituents of a picture become clear, especially when focussing on the spatial issues. "2.0271 Der Gegenstand ist das Feste, Bestehende; die Konfiguration ist das Wechselnde, Unbeständige." 26 The objects are there, showing as how they are. The coordination of them makes different decisions of arrangements. These arrangement can have make possibilities. Commonly we arrange an altar and the audience facing each other for having better communication. However, picturing as the other proposition we talked before that 'The audience stands in front of an altar.' is not impossible, but we have a difference as the former one in mind though. Instead of referring to the seat, 'audience' is now related to the persons, since it 'stands'. It could also be read in another way that they stand in a line, the audience has its back towards the altar, or we can even manage them as 'The audience hangs above the altar.' in a totally another coordination.

These statements are all possible, possible to picture. They are not false, as long as the altar is still the place for the religious performance and the audience is listening or viewing. However, in these unconventional way of arrangement, that an altar can still function as the altar for those hanging audience, or if they can listen and view the altar from their hanging situation, will decide whether this space really functions as, e.g. a chapel.

Seeing from the other angle, whether a space functions, on the one hand, depends on if well organised, however on the other hand, essentially relies on whether the elements work as they are supposed to. If every object is originally colourless and born with its perceivable form, then the question emerges, which essence actually makes one object in reality be named as - or related to - this element in our picture, where we call it an 'altar', when we actually see one? ('perceive' one to be more accurate)
FORM / FUNCTION

NAMES

are presented by those symbols, appearing in propositional expressions, where we have the pictorial objects described. However, an object is not fixed to its certain name(s) as we call as we just see it. They should be considered as two individuals. We use sentences to describe those what we think of, and we have the objects from our pictures corresponding to the names in our propositions. “3.3 Nur der Satz hat Sinn;” the sense, which we speak out of, can only exist within the proposition, where we have a picture mirrored. In this case, “nur im Zusammenhang des Satzes hat ein Name Bedeutung” whenever we find our picture a proposition to respond to, whenever the pictorial objects are related to the expressional names, and the names can stick to the objects only in the expressions. “3.203 Der Name bedeutet den Gegenstand. Der Gegenstand ist seine Bedeutung”

At the first look at the definition of an ‘altar’, we regarded the more important meaning of it, was the religious performance, however, there is, in fact, with its geometric factor that we consider this certain object that we have pictured as an ‘altar’: an elevated structure.

2.0141 Die Möglichkeit seines [vom Gegenstand] Vorkommens in Sachverhalten, ist die Form des Gegenstandes.” This certain geometric necessity makes a object - which is suitable for this dimension - able to have the opportunity of being used as an ‘altar’, where the religious performance can take place, to form an atomic fact. With every object being coordinated in the suitable order and having the suitable dimension for its functionality, then the thought we conceive can therefore find its suitable form.

Eventually, the names become irrelevant, it depends only on how we do with the objects around us. We shall name an object, only when it functions as it is about to be named. However, why can we call an object ‘car’, which carries people from one place to another and probably has four wheels, four seats inside, and a running motor, before we even drive it? Actually, when the name ‘car’ is mentioned, its description as above has also been implied and thought of. This structures a ‘car’ with a coordination related to how its components are built up and collaborating. The fact, that the car can drive, is embedded within this car-object as its potential to function as one. “2.0123 Wenn ich den Gegenstand kenne, so kenne ich auch sämtliche Möglichkeiten seines Vorkommens in Sachverhalten. (Jede solche Möglichkeit muss in der Natur des Gegenstandes liegen.)”

Functions have shown in two different forms: the functionality of an object and the function of a space. The same object could be called differently if we use it differently; the same space can function diversely if we arrange for it in distinctive ways. Take a table for example. The name is usually referring to a level at 85 cm high which provides the possibility for us to set up a dinner, a workstation or a conference. In these cases, chair usually comes along with it. At level 50 cm high, a chair functions not only as a seat, but also provides a back for leaning on. Chair offers comfort, while we have dinner, work or conduct a discussion on the table.

However, a usually seen behavior would be that a person, while discussing, stands up, walkabout, and finally end up half sitting on the table instead of on his or her chair, and keep on the conversation. Do we call the table now a chair, while the others are still working on it?

How about naming the spatial functions as we usually do? Imagine a living room. As we usually call it, a few objects in it come up: a sofa, a coffee table with a lamp and probably a TV set. They stay in a certain topological relation to each other, where the coffee table stands next to a sofa, which faces towards a TV set. We would probably describe a living room as this way.

Nonetheless, is a ‘living room’ the only spatial function having these objects? How about a hotel lobby? A hotel lobby has all these objects as well. Moreover, how about a waiting room in the hospital? Furthermore, how about their building names, which we just used above. With the beds they have within, either a hotel or a hospital, what makes them so essentially different from a place we call home?

Can the name of a function, along with a bunch of the names of objects, really sufficiently say what a function really is?

3.142 Nur Tatsachen können einen Sinn ausdrücken, eine Klasse von Namen kann es nicht.” What makes actually a ‘living room’ a living room distinctive from the other spatial functions? From a lobby or a waiting room, for example? The great difference exists within the activities that we conduct, and for whose fulfillment, spatially, lies within the coordination of the objects.

All in all, it is its dimensions which gives it the potential to function, a pictorial object could correspond to the name in our proposition; it is the coordination of its objects which makes this certain activity happen within, the certain space could then be named as the function as we heard.

Geometrical dimensions, along with topological coordinations, are the forms for functional activities.
In the sense of space, how a human being perceives and interacts with his or her surroundings, builds the true meaning of architecture. We describe of a feeling in reaction to our environment with languages, where the configurations of words, names corresponding to the objects in reality, make our expressions understandable.

Each object has its own properties and potential functions, properly to form the described picture that we have in mind. Each interaction between us and one object or objects constructs a micro-universe, which exists within a macro-universe, where all these sub-systems build up together. There are usually names for them, living room, dining room, staircase, toilet or bedroom, and so on, which compose an even higher level in the system, such as an apartment, a house, a hotel or hospital.

An architecture is built up not only by each brick of it, but, more essentially, by every activity that human conducts within.

Thoughts of this section are inspired from the following texts, books:

"Tractatus logico-philosophicus"
- by Ludwig Wittgenstein, 1921
- English translation by C. K. Ogden

Keywords:
Object, Property, Function, Activity, Movement, Sub-system
**ACTIVITIES**

vitalise architecture. They endow the meanings to the built. If we do not look out through a window, where are the joys of having one?

Architecture is the creation from civilisation, whose goal is meant to assist human for their needs. From a very simple shelter, providing protection from danger or rain, to spiritual temples or churches, until high-tech factories or surgical hospitals, the intentions are one: to make our activities possible.

*Spaces* are the places we conduct our activities within an architecture. Either a surgical operation is on going, a radio broadcast is on air, a ceremony is being carried out or a movement from one room through the corridor, via a staircase to the another room upstairs. Either different objects are utilised in a similar way, or one object is used however in many ways and have endowed many names.

Chairs and tables might come handy when the radio is being broadcasted along with microphones and headsets, which could also be needed when a Mass is being hold in a church. An operating table is definitely needed for a surgeon. However, the very same table does not seem to be a one for the patient, who, at the mean time, lies on it. For this patient, it is not table but an operating bed.

The same activities can, in fact, be conducted with different objects. An emergent surgical operation could borrow a few tables from a broadcasting room; the audience might sit on steps if the auditorium is over-crowded. A table for operation might be able to be at a suitable level for a priest to rest the bible and start to preach. These potentials of objects shows the possibilities are embeded within their geometries, just about whether they are suitable to our activities.

The interaction between objects and humans runs in two directions. Not only the objects have to have the certain geometry to match our utilisation, but our bodies must also be at the matching positions.

It is clear, if the hands of the surgeon fail to reach the operating table, the operation could never be carried out; no step forward to reach the object, one can never reach upstairs, even if it is an escalator or a lift.

These common named objects as we know, tables for dining, chairs for sitting ..., etc., have their common forms, which have been improved with time in order to better function for the activities. They represent our using experiences in *ergonomics*. However, if we leave those customed name and form behind, we see clearly, their dimensions are the only thing needed.

> “3.144 Sachlagen kann man beschreiben, nicht benennen.”

34 States of affairs can be described but not named.

of objects around us help at our activities. They are the true meanings of objects. Without the functionality, the objects cannot interact with our activities. Functionality provides an object the possibility to be the one that we call it.

In the term of space, functionality simply means those usage that we conduct with or within the surroundings. When we are making notes in our classroom, our school tables provide its functionality to have our notebook and our arms resting on it; if a patient is scheduled to have an operation, an operating table is in need, for the patient to be treated and for the surgeon to fulfill his performance. When the students are learning, the surgeons are operating or the patients are being operated, these *table* objects are then functioning.

The classrooms, where many school tables are gathered within, would help the students learning and the teachers teaching, as it should function, as long as the teachers and students, along with their tables or podiums being used, are performing; so does a surgical operating room work as well. The necessary parts in a operation, the surgeons, the patients, the operating tables... etc., perform in the room, which functions as an *operating room*, with the lights on, written *in surgery*, on the door.

Wherever the teaching and learning happen, there is a school. We might not need a room for education, but the rooms, which function as classrooms or administrations, can compose a school. For the humanitarian organisation “Doctors without borders”, the
Properties of an object decide how it could function. Of a car object, those component objects, which it needs to function as a car - e.g. to run or to transport - are its properties. Properties are those necessary ones, through which the object is understood.

Properties could be the attributes, such as size, with which a car, for example, has the capacity to transform; they could also be the components, such as a motor or wheels for a car to run, or an operation room for surgical center to treat the patients.

Speaking of space, geometrical properties show strongly an influence on how an activity could interact with the objects. Collaborating with human movement, an activity can find a comfortable geometrical property on an object, where it can operate, say, to land its hand on.

Similarly called as table, these objects perform one same functionality to achieve the human needs: providing an elevated level, high enough, for our working items to rest on: a set of computer workstations, a set of silverware for dining or a patient waiting to be treated. Though their slightly different operations, they could also be given names: a study table, a dining table or an operating table, their basic property - having a higher level - categorises them into a same group.

Surely, not in an emergency would we let any doctor operate our bodies just on some unsterilised pool table - certainly it is debatable whether the surgical operation would work, even if the pool table is sterilised, it is just a metaphorical expression here - however, a pool table has the possibility to function as the one for a surgical purpose, and so would the other objects such as a dining table or a study table - but not a 'time table', where the object named with this symbol 'table' here has no property as a higher level, and this symbol signifies differently as the table in the sense with elevation. (The two different 'table' significations can however come from a similar sense, as where you put things altogether, such as pool balls, dishes or train schedules, multiplications.)

Properties, such as the height of a table, the cleanliness of the pool table, decide if the certain activity - in this case, surgery - would take place involving this object. Regarding architecture, the geometrical properties would be the main issue here to focus on. However, object properties do not just come alone for this issue. An object also has to interact with the others, in order to activate its surrounding space.

In order to know an object, I must know not its external but all its internal qualities.

35 In order to know an object, I must know not its external but all its internal qualities.

determine the relation between objects in space. If not only represents the arrangement in those pictures, but also corresponds to the order in the propositions that we thought of. Intepreted into a coordinate system, the geometrical property of objects, and their coordinations can be measurably represented.

Coordination of objects composes a condition, which allows the activity, related to these objects, to happen. However, whether the coordination is properly managed, crucially effects the chance if the activity works well or should takes place here.

In a proposition, the coordination object is described as its predication 'stands in front of'. This relates the two objects, the one we call 'altar' and the another 'audience'. Correspondingly to the picture we have in mind, we locate these two objects in positions. The positions, following the propositions, are only relevant to each other. Though the names characterise the objects, and the whole situation is implied, but the more information is needed, in order to form a whole picture - as a fact.

More coodination information could be given to confirm us more details in the picture, such as regarding whether the 'altar' and 'audience' are facing each other. It can inform us about the surroundings as well. Such as how the distance is from the chord to the altar, or from which direction the people come in... etc. With all the information together, we might then refer the building object as a church in other propositions, but not the another way around.

The coordinations are not just in the sense of the topology between objects, but between the components of objects as well. How the legs of a table are situated in relation to the weight center of the whole table, effects directly the stability of this object, which influences the liability to be used in a surgical operation. So is the coordination of the columns, if it cannot help a building bear its own weight, then the rooms within vanish, no activity can take place inside, and the building object cannot be called an architecture.

We used geometric coordinates to describe the coordinations. In the world coodination system, the positions of objects are commonly expressed as points with (x, y, z) coordinates, along with a parenthesis to present as a set, and the relations between objects are presented by imaginary vectors. "[...] a vector is a direction and length [...]. Mathematically, a vector is an abstract object that is a part of vector space, itself a mathematical object." Correspondingly, we can also find this mirroring in Wittgenstein's philosophy in the sense within propositions,"3.144 [...] (Names resemble points; propositions resemble arrows, they have sense.)"

Either between objects or within an object, coordinations forms the orders of a system with different levels.

37 [...] (Names resemble points; propositions resemble arrows, they have sense.)
describes a level system like this one, mentioned from above. Within each of the objects, there is a micro-system, and the interactions between each of the objects at the same level compose a macro-system.

Human body conducts activities with its parts. Movements of maneuvering the body parts change the body into many possible positions. The position possibilities rely on the different coordinations of the limbs and torso. By observing a micro-system, an arm can be analysed as a system composed by upper and lower arms along with a hand, whose coordination would be illustrated as poly-lines connecting each limb joint. The same observation works throughout each of the body parts. The study is called anatomy. The analysis can goes further deeper towards atoms.

These combination of the position of our body limbs varies in different way, according to our interaction with surroundings. Spatially, the directly contacted would be the furniture pieces.

Within one piece of furniture, a micro-system - familiar to the one in human bodies - could also been seen within, and its analysis is essential for furniture design. The related study, “of people and their working conditions, especially done in order to improve effectiveness”34 is called ergonomics. A chair, “a piece of furniture with a raised surface used to sit on”38, is probably the one kind of furniture that a person spends the longest time with, in the whole life. A chair varies its form according to the different functionalities at needs: bar stools at bars, seats are those fixed in trains or theaters, airplane seats fly with us, car seats drive with us, so does bicycle saddles ride with us and wheelchairs are designed for the handicapped. The forms can also be altered or improved in agreement with the ergonomical studies for creating better working efficiency or keeping the body in a natural, relaxing position in order to be comfortable during the long time activities. The various forms of a chair are the results of furniture designs, a combination of ergonomic and esthetics.

Many pieces of furniture together compose a space we live in. The study of the arrangement between furniture pieces and in the relation to human bodies is interior design. At this level of folding-systems, the pieces of furniture can be related to the philosophical objects - their size, dimensions, colors are the properties, and each of them has potetials to function and be named as how we usually utilise them. The coordination of furniture pieces works together with the determination of their possible functionality.

Between the furniture pieces, the coordination, e.g. the distance, determines a way how we would approach them, and indirectly effects how their geometry could alter to function better. In one’s den, for example, the distances between a table, a chair and a book shelf influence the body movements towards each of them or inbetween them. Commonly, we sit on chair and working at the table, and if we need books from the shelf, we stand up, walk towards it, stand in front of it and start searching. The movements are sitting, walking and standing, separately. However, if the table and the shelf are close enough, we could just extend our bodies and reach directly from one to the other, then we change our usual way of approaching the shelf without walking towards it; in this case, we might also wish the shelf become shorter, for sparing us from standing up. Although, even if they are still located far from each other, however, we might be able to stay sitting on the chair to reach either of them, if the chair additionally has wheels and the possible function of transporting. The movement involved here is only sitting left.

The ’programming’ of spatial functions in a building is an important part of architecture designs. As this sense of ’programming’, as Peter Eisenman once the word coined, refers to the generating of possible combinations of spatial functions in a building, in order to find the suitable solution from many distinctive design aspects, such as convenience, efficiency, structure, or indirectly from aesthetics. This process is usually proceeded in our mind, based on our experiences and visualised with some trying drawings. In a topological arranging method we intend to move those bubbles, which represent rooms with functions, and to change those links between them, which mark the accessibility from room to room, in order to reach optimisations. The room bubbles have their properties, such as their sizes, their potential functions and their coordination, represented by the whole topological illustration.

This level-to-level-folds indicate a hierarchical system. We might see it as many folds including objects, and the objects from different levels do not seem to correlate. However, through Wittgenstein’s philosophical point of view, this hierarchy looks more like a chain reaction between the levels. As colourless as an object is, whether an architectural object could be named a surgical clinic, depends on if it is equipped with a room, where an operation could be performed and a table is installed, on which the hands of a surgeon could hold a scalpel to treat a wound of the patient.

The whole scenario would be totally seen differently, if the operating hands are of a cook and the operated one is a chunk of pork. The whole chain has been then altered, a cooking table and a kitchen are coming into play and the architecture piece is called ’restaurant’.

Certainly, only a cook and a knife can do probably not much, not for running the whole restaurant. For a functioning architecture, each part of it has to work properly, and then the building can mean what its name is.

Observing from top side down, the dimension matters, of objects and of their coordination; influencing from bottom up, changing just a few centimetres higher could alter the definition of a building. Could the various forms of objects be integrated into one but many, to perform the necessity of dimensions? Could the dimensions be flowing as human bodies change between positions in a continuity?

What then, if the various functionalities and geometries permeate into each other?
A similarity between philosophy of Wittgenstein and structure of Object-oriented Programming is embedded in the parameterised geometry. We are moving from a ‘computerisation’ era, when we were learning and using computer as a drawing tool, towards the era of ‘computation’, a term meaning “the use of the computer to process information through an understood model, which can be expressed as an algorithm.” The algorithmic influence will move further, deeper into the design process. Computer is not just a tool. It becomes the design rationale.

Think algorithm!

---


Thoughts of this section are inspired from the following books:
- “Architectural Design Magazine”
- Volume 83, Issue 2, “Computation Works”
- “Elements of Parametric Design”
- by Robert Woodbury, 2010

Keywords: OOP, Python, NumPy, RhinoCommon, Grasshopper, Parametric Design

Computer technology has harvested its popularity increasingly over last two decades. At the beginning of its revolutionary era, the computer users struggled to be familiar with this new tool as much as possible, and tried to manage our usual paper works digitally. At that time being, people attended classes for certificates proving their abilities of in software skills, in order to follow up the trend. These softwares to learn were made by computer scientists, the experts at writing software scripts for living. Twenty years later, with the open-source information and the online forums, serving as utopian knowledge exchanging platforms, everyone can learn freely and eventually do more or less programming or just modification. It may just be a few lines in an HTML of one’s blogs or webpages, statements in Actionscript or in Processing for a graphical presentation, or in a small widget in the beloved smartphone for making the daily life more efficient.

Gradually, learning to speak one of all the programming languages becomes gradually inevitable, so happens in architectural practice as well.

Observing in one small scope of the development in the user’s experience on one common, long-time software utilisation in architectural practice: AutoCAD, from Autodesk. On its early versions, around the periods when the R12, R14 versions were released, an addition of a drawing pad was a trend in the offices. Simulated to a real sketch board, the drawing pad not only having pre-set commands written on board, but coming also with a pointing pen, allowed users to draw in a virtual world as well as on a physical table, and for the other users without pointing pens, the simulation was pressing on screen buttons with a mouse cursor but without physically holding a pen. These simulations of the reality, transformed the drawing habits from paper-based to monitor-based. It had saved much of time-consuming tasks, such as correcting errors, if one wrong ink stroke fell on tracing paper and a scratching correction had been suggested, or reproducing a massive amount of one same element.

However, following the parametric current, dynamical blocks have been introduced into AutoCAD 2006, with which, customised parameters could be set up within the block. Then the duplicates of this block can be uniquely modified by using the pre-set parameters, but still stay in the prototyped form as the original. Additionally, the parameters can also constrained by each other. For example, the copies of a window block can have different widths by stretching parameter, different sliding directions by mirroring, or angles of opened window panes by rotating, or, an axis of pane rotation stands constrained in the same proportions to both ends, no matter how much the window width has been stretched.

This was the development of parametricism in a form of virtual drawing. Abstracter would it be the use of a virtualised programming language - such as Grasshopper for Rhinoceros - where the constrains or parameters are not manipulate in forms but by giving numbers, and the commands are not to be called by pressing buttons but are connected through pipes to build a workflow. Then, even more abstract - scripting texts writing in softwares, such as in Lisp for AutoCAD, in Java Sctipt for Processing, or in Python for Rhinoceros.

Along with the popularity growth of parametricism, the level in design process has been influenced deeper by algorithm. “Four or five years ago, excitement centred largely on the emergence of new fabrication and manufacturing techniques [...] the onus on architects lies in the application of computation ‘to simulate building performance, to incorporate performance analysis and knowledge about material, tectonics and parameters of production machinery in design drawings’. Just as revealed in the editorial of the issue “Computational Works”, the role that algorithm plays in architectural design has been further changed “from being merely a generative tool to being at the heart of building and practice, and even beyond to the monitoring and assessment of completed structures.”46 Computationalisation did not just merely speed up our working process, but also led us to a whole new level of thinking progress, about what the truly content is beneath all those fancy appearances.

Through Wittgenstein’s eyes, coordinates are not just merely indicate the locations, and the relation between the coordinates along with the geometry by them is not just arbitrary, but as what Hugh Whitehead once described as “embedded rationale”, where the geometry is armed with its downstream logic, and finds an efficient path to manage geometrical relationships upstreams, whereby the coordination in ex-Wittgenstein’s extended picture explanation is just as “in the modelling of complex topological spaces, while all of the local operations are Euclidean, it is how these local operations are glued together that contains the new topological relationships and what must now be incorporated into the CAD software.”47 So as “Fabian Scheurer [...] argues that design is all about decisions and that delinking these to an algorithm always means following predefined paths. [...] Through the creation of new tools, new ways of thinking and new solutions can be found.”48 Based on the Object-oriented Programming, with language expressions and embedded structural geometry hierarchy, the flow of scripts works along with functions, all in all, either those built-in in softwares and programming languages or those user-customised scripts, in order to create the pictures as we have in mind.

41 http://en.wikipedia.org/wiki/ActionScript
42 http://processing.org/
47 “Inside Smartgeometry”, Brady Peters & Terri Peters, p.15
Object-oriented programming, "is a programming paradigm that represents concepts as "objects" that have data fields (attributes that describe the object) and associated procedures known as methods." OOP represents a structure, that each objects has properties, such as color, length, width, or coordinates, which describe what or where this object is, and functions, how this object could produce or be utilised. In the sense of daily life, speaking of a car for example, its general properties would be its color, length and weight and its four wheels, which form its class category, if we speak specifically of a certain model, say a Mini Cooper, what is made to be its properties would be the equipped four wheels and also the specifications, the body painted in red, a black convertible roof, with 3.3 meters length and 686 kilograms. And it has a potential to function: drive.

In the other senses, speaking of space, chairs form a class having some general properties in common, such as a needed seat board with around 45 to 55 centimetres in depth and width, and supporting legs which hold this seat stay at a certain height. As to which color it is, or if a lean is installed, more specifications about the different types of chairs can be further given, as long as it serves the users as a possibility to sit. Additionally, the difference between a door class and a window class is not just defined by the size of them, but the functions that, generally, doors are provided for passing through, and windows are however for seeing through, however they can probably be categorised into the same super-class though - a class higher - a class of holes, a way of connecting two sides physically.

Using a language - either a programming language, such as Python, or a language that I am using for writing at the moment - is a way of communication by expressions from one side to another. However, in the case of programming, the expressions would then be compiled by a processor and the facial expression won't count. No matter which kind of the programming language it is - object-oriented or not - if the sentences are not written down and do not syntactically understandable, the communication cannot happen, and the program will not function. That is the way how Wittgenstein explains the world in that little red book "Tractatus" that "1.1 Die Welt ist die Gesamtheit der Tatsachen, nicht der Dinge." So is it in the object-oriented programming. Although the structure of the database has been built into a objects/classes structure, and their functions are included into their underneath sub-layer properties, for each script, there must however be an environment, where the global parameters are given to those objects which are operated, and where the whole mechanism can run towards a final goal. The goal would be - in the sense of design - to generate a final geometry.

"Software Development Kit (SDK)" of Rhinoceros, the term "Common" in it is "meant to be just that; an SDK that can be used across Rhino platforms." The plug-ins created by the users with RhinoCommon can be used without further alternation in different operation systems. Via this SDK the users have a way of speaking to the core of Rhinoceros. Rhinoceros 5 on Windows System (so does RhinoOSX on Mac OS) has a channel for the Python programmers called the .NET Framework (respectively on the Mono, a cross-platform for Mac users), where the programmers write in Python to utilise the SDK objects.

By looking into the structure of the SDK, we can clearly see an application of object-oriented structure. Point, line or curve and surface or Brep (Boundary representation) surface, for a common sense example, are the classes under the geometry namespace. Having the x, y, z coordinates as its location, a point instance - as an object initialised from a point class - can then be defined in the world coordinate system. However, a point can be also derived from a line or curve instances by giving the t parameter when using the "PointAt t" function under their instances. A point can also be found on a surface or brep by calling "PointAt u,v" function with the given u and v parameters. From these two methods, we obtain the point's information with its world coordinates, but we also notice, this t’ or ‘u, v’ parameters are the "coordinates" within the corresponding curve and surface systems. For on a line or curve, it becomes single dimensional, even a curve goes with more degrees. Also for on a surface, it only goes with (u ‘+1 ‘+ v’) vectors, where we see i and j as two directions on surface, similar to the x and y vectors in world coordination system.

This reveals not only an order as a "folding system" (as mentioned in assay 2, folding system) in the RhinoCommon SDK and in geometry, but the processes of "initialising", "property endowing" or "functioning" also correspond to Wittgenstein philosophically.

It does not matter either the points are created by giving directly the coordinates or derived from the other geometries with their positioning information relative to the originating geometries. All points belong to the Point class, thus each of them has a set of indicating coordinates relating to the world origin (0,0,0). With this relation, an environment system can be built and the geometry can then be defined in space. The point-generating function is not a function of points, therefore belongs to the form which they originate from. Under the Curve or Surface classes (lines belongs to a form of curves and breps a form of surfaces), there is a same "PointAt" function by default, however, due to their different dimensional properties, this function works differently. Which also defines how these two geometries are distinctly composed: either with one dimensional parameter ‘t’ on curves or two ‘u, v’ on surfaces.

The world is the totality of facts, not of things.
from surfaces, they all - so does the Point class - are sub-classes under the super-class “GeometryBase” and inherit its properties and functions, such as Transform() or IsDeformable.

Furthermore, another example of cross inherittance of properties and functions between two parallel classes can be found within the relationship between Surface class and Brep class.

“Boundary Representation (BRep)” surface is “a surface or polysurface along with trim curve information.”54 Within a BRep instance we can find a properties called “Faces”. By using “Faces” we can retrieve the surface(s) information within this representative. These faces, composing a brep geometry, are instances from Surface class, ergo the properties and functions applied for the Surface class can also be applied for them, i.e. for the brep instance from Brep class. In other words, if we want to get the points on this brep surface, by applying “PointAt” function, however not directly on the brep surface but on the faces, which it is composed of.

Arrays

“systematic arrangement[s] of objects, usually in rows and columns”55, equipped as a class in n-dimension in NumPy, “a fundamental package for scientific computing with Python”56, extend the ability of modelling by writing Python scripts in Rhinoceros 5.

In many cases the manipulations of the n-dimensional arrays (nd-array) help to speed up the calculation and shorten the redundancy of scripting text. Converting nd-arrays to either lists and tuples in Python language or organising into dictionaries, which are “sometimes found in other languages as ‘associate memories’ or ‘associative arrays’”. Unlike sequences, a type of data storage, including arrays, lists and tuples, “which are indexed by a range of numbers, dictionaries are indexed by keys.”57 Meaning, in sequence storage, data are retrieved by given index number, comparing to the data in dictionary, which are found by given index names.

With the help from the plugin “Grasshopper Python”, “the Python interpreter component for Grasshopper that allows to execute dynamic scripts of any type”58, combining with the tree structure in Grasshopper, algorithm modelling can be easily, on the one hand, comprehended on the visualised programming canvas, and, on the other hand, clearly organised in classes categories within Python and faster calculated with NumPy arrays.

For example, imagine a grid of points in the world coordinate system, where they have three coordinates, representing their relations to the origin zero point as in x, y, z direction. In case we want to operate their coordinates differently however in a similar manner, such as adding values to each of their z coordinate according to their x and y coordinates by each result of $\cos(x)\sin(y)$.

By using lists, the three coordinates can be put into layers in a oder: firstly in 'n' amount of columns, then in 'n' amount rows. The total amount of coordinates is $m \times n \times 3$, and the data stand on the first position (index 0) are the points with the same x-coordinate. However, if we intend to covert the list to a flipped oder, as we would like the first data (index 0) to be the first row in grid, we have to relist them all.

Luckily, by converting this list into an array, the reordering becomes simpler. The array is now in shape (m, n, 3), in an oder as the same as the list. However, when retrieving the first row of points, we transpose the array in exchanging the first and second di- mensions, i.e. making the shape into (n, m, 3). In this order, the first position of the array will be the points we want. Furthermore, transposing the dimensional oder into (2, 1, 0), we will have a shape (3, m, n) and an array with all the x, y, z coordinates seperately: all x-coordinates on the first, y one the second and the z the last position.

The final step to calculate each $\cos(x)\sin(y)$ then add to each z-coordinate seems readable and manageable, and the task is fulfilled.

Polynomials

can be expressed mathematically in a relation between values $(x, y)$:

$$y = f(x) = a_0 x^n + \cdots + a_k x^k$$

which, if drawn on a x-y 2d-plane, reveals the relation between x and y coordinates of all the points composing the polynomial curve with a degree of $n$. The constants $(a, b, ..., c, d)$ are called coefficients and the ‘x’ is called variable, by giving different x values we find the corresponding y values on the polynomial curve.

Each geometry we have modelled in 3-dimensional software has mathematical equations behind. Each graphic we saw on the computer monitors is the calculation result of the 2d-projection of the modell we build. Behind a curve, we find one or more corresponding polynomials.

A function call “polyfit”59 in NumPy, helps us to find the coefficients of the known polynomial curve. By giving sets of x and y-coordinates of composing points, the function returns the coefficients according to a second given parameter - the curve degree-60. This results a approximately curve which goes by each point we have input.

Sequentially, applying the returned coefficients into function “poly1d”, we create a mathematical polynomial equation, where, with further inputting x-values, we obtain the corresponding y-values from the

---

54 see Brep Class in http://4.rhino3d.com/5/rhinocommon/
55 http://en.wikipedia.org/wiki/Array
56 http://en.wikipedia.org/wiki/NumPy
57 http://docs.python.org/2/tutorial/datastructures.html#dictionaries
58 http://www.food4rhino.com//project/ghpython
60 http://en.wikipedia.org/wiki/Curve_fitting
function returns. This forms a planer curve based on a x-y relation. If we have an extra relative information either on x-z or on y-z plane, a three dimensional curve could be drawn by interpolating its points in x, y, z co-ordinates.

A three dimensional curve is shaped by the two projected polynomials. When a polynomial is given, by determining a domain on x-axis, we can retrieve the y values from the poly1d of xy-projection, and respectively the z values from xz-projection. In this way, the approximate curve can be drawn point by point.

If we equally segment two different curves into divisions and connect each dividing point from each side with straight lines, we find approximately a more smooth surface if the amount of segments arises. When the connecting lines are infinite, we can imagine a whole surface. This creates a 'loft surface', which belong to the Surface class as a function to generate surface in Rhino. Between these two generating curves, we found 'tween curves', which are fallen on the lofted surface and can be created through function 'CreateTweenCurves' in Curve class. Each of the tween curves has a representative of 'poly1d' in NumPy, whose coefficients are dividingly in proportion between the two ending curves. In other words, the coefficients of ending polynomial curves can be stored in a array, after being divided into segmental values and additionally inserted into the array, the information of a certain point on a certain polynomial curve can be accessed easily through array manipulations.

**Examples**

**Gyroid** minimal surface in mathematical expression:

\[ \sin(x)\cos(y) + \sin(y)\cos(z) + \sin(z)\cos(x) = 0 \]

[61](http://mathworld.wolfram.com/Gyroid.html)

Loft surface from polynomial curves, as mathematical expressed in:

\[ y = f(x) = a x^n + b x^{n-1} + \ldots + c x + d \]
WORLD SYSTEM

contains the points, as shown in illustrations, with the coordinates in relation to the origin of the system. The points of a tessellation pattern have equal distances to the adjacent ones (Figure E3.1.1), as in the usually case. If the alteration of the points’ x-coordinate as each value of them is re-calculated as the cosine value of it, the distances between points in x-direction are changed to a repeating proportions, as cosine values waving between 1 and -1. Similar order will also happen and form waves in y-direction if we have the y-coordinate change to its sine value. If we then have the z-coordinate relate to the x and y-coordinates of the points as in \( z = \cos(x) + \sin(y) \), then the tessellated geometry we have, is an approximate of the surface representative of the equation in form.

**Example 1.1**
- triangulation
- coordinates in world system
- point \((u,v,0)\) add \((f(u), f(v), f(n))\)

<table>
<thead>
<tr>
<th>(f(x))</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>(f(y))</td>
<td>0</td>
</tr>
<tr>
<td>(f(z))</td>
<td>0</td>
</tr>
</tbody>
</table>

**Example 1.2**
- triangulation
- coordinates in world system

<table>
<thead>
<tr>
<th>(f(x))</th>
<th>(\cos(u))</th>
</tr>
</thead>
<tbody>
<tr>
<td>(f(y))</td>
<td>(\cos(u))</td>
</tr>
<tr>
<td>(f(z))</td>
<td>(\cos(u))</td>
</tr>
</tbody>
</table>

**Example 1.3**
- triangulation
- coordinates in world system

<table>
<thead>
<tr>
<th>(f(x))</th>
<th>(\cos(u))</th>
</tr>
</thead>
<tbody>
<tr>
<td>(f(y))</td>
<td>(\sin(v))</td>
</tr>
<tr>
<td>(f(z))</td>
<td>(\cos(u))</td>
</tr>
</tbody>
</table>

**Parameters**

- \(d_u\): differential domain in x-axis
- \(d_v\): differential domain in y-axis
- \(d_n\): differential domain in z-axis
- \(f(u)\): math function in x
- \(f(v)\): math function in y
- \(f(n)\): math function in z

- \(\alpha\): differential domain in x-axis
- \(\beta\): differential domain in y-axis
- \(\gamma\): differential domain in z-axis
- \(\mathcal{f}\): math function in x
- \(\mathcal{g}\): math function in y
- \(\mathcal{h}\): math function in z

- \(\theta\): differential domain in x-axis
- \(\phi\): differential domain in y-axis
- \(\psi\): differential domain in z-axis
- \(\mathcal{J}\): math function in x
- \(\mathcal{K}\): math function in y
- \(\mathcal{L}\): math function in z
**Example 1.4**
- triangulation
- coordinates in world system
- hyperboloid geometry

\[
\begin{align*}
  f(u) &= 0 \\
  f(v) &= 0 \\
  f(n) &= u*v
\end{align*}
\]

**Example 1.5**
- triangulation
- coordinates in world system
- point \((u,v,0)\) add \((f(u), f(v), f(n))\)

\[
\begin{align*}
  f(u) &= \cos(u) \\
  f(v) &= \sin(v) \\
  f(n) &= u*v*\cos(u)
\end{align*}
\]

**Example 1.6**
- triangulation
- coordinates in world system

\[
\begin{align*}
  f(u) &= \cos(u) \\
  f(v) &= \sin(v) \\
  f(n) &= u*v*\cos(u)*\sin(v)
\end{align*}
\]

**Example 1.7**
- triangulation
- coordinates in world system

\[
\begin{align*}
  f(u) &= \cos(u) \\
  f(v) &= \sin(v) \\
  f(n) &= u*v*(\cos(u)+\sin(v))
\end{align*}
\]
SURFACE UV SYSTEM

creates an new coordination environment of the points on the surface, for each points with the (u, v) values, the topology of them stays in a same way as they are in world system, and the u and v values are seen as the x and y but not along a straight direction but along the curvy edges of the surface. This could be imagined as the points are sticked to the flat world system surface and the surface has been bent and stretched, but, to the environment surface, the points stay on the same position and have not moved.

Additionally, adding the sine and cosine alterations to the point coordinates, has even more intriguing results.

example 2.1
• triangulation
• coordinates in world system
• point(u,v,0) add (f(u), f(v), f(n))

\[ f(u) = 0 \]
\[ f(v) = 0 \]
\[ f(n) = 0 \]

example 2.2
• triangulation
• coordinates in world system
• point(u,v,0) add (f(u), f(v), f(n))

\[ f(u) = \cos(u) \]
\[ f(v) = \sin(v) \]
\[ f(n) = \cos(u) \]

example 2.3
• triangulation
• coordinates in world system
• point(u,v,0) add (f(u), f(v), f(n))

\[ f(u) = \cos(u) \]
\[ f(v) = \sin(v) \]
\[ f(n) = \cos(u) \sin(v) \]
example 2.4
- triangulation
- coordinates in world system
- point(u,v,0) add \((f(u), f(v), f(n))\)

\[
\begin{align*}
  f(u) &= \theta \\
  f(v) &= 0 \\
  f(n) &= u*v
\end{align*}
\]

example 2.5
- triangulation
- coordinates in world system
- point(u,v,0) add \((f(u), f(v), f(n))\)

\[
\begin{align*}
  f(u) &= \cos(u) \\
  f(v) &= \sin(v) \\
  f(n) &= u*v*cos(u)
\end{align*}
\]

example 2.6
- triangulation
- coordinates in world system
- point(u,v,0) add \((f(u), f(v), f(n))\)

\[
\begin{align*}
  f(u) &= \cos(u) \\
  f(v) &= \sin(v) \\
  f(n) &= u*v*cos(u)*sin(v)
\end{align*}
\]

example 2.7
- triangulation
- coordinates in world system
- point(u,v,0) add \((f(u), f(v), f(n))\)

\[
\begin{align*}
  f(u) &= \cos(u) \\
  f(v) &= \sin(v) \\
  f(n) &= u*v*cos(u)*sin(v)
\end{align*}
\]
shown by an example, taking a partial basic surface of Gyroid surface, which, in its translated, rotated or mirrored form, is identical to the other parts of the whole periodic surface, this partial edge (in green) shows the original surface as geometrical representative of the equation: \[ \sin(x)\cos(y)+\sin(y)\cos(z)+\sin(z)\cos(x) = 0 \]

If the surface has been put into an environment force system, shown as the outer frames, the composing points on its outlines, along the normal vector to the base plane, according to the heights, are dragged toward a combination vector direction by randomly chosen either cosine or sine values of its height, as in \( z \), in \( x \) or \( y \) axis direction, for instance: \( y = a \cos(z) + b \). Where we then obtain a slightly deformed partial basic facet of the Gyroid surface.

**Example 3.1**

- Geometry, built in one coordinate system, affected by an outer system

\[

\begin{align*}
\text{f}(x,y,z) &= \sin(x)\cos(y) + \sin(y)\cos(z) + \sin(z)\cos(x) = 0 \\
\text{f}(x) &= \sin(a \cos(z) + b) + c
\end{align*}
\]

- World coordinates: original, unaffected minimal surface Gyroid shown in outline curves (red), based on equation:

\[ \sin(x)\cos(y)+\sin(y)\cos(z)+\sin(z)\cos(x) = 0 \]

- Surrounding framing system:
  - given frame size in width(W),
  - height(H),
  - depth(D),
  - amount(A),
  - thickness(T)

- each of frame corners (in four sectors) is extended by an addition of:

\[ \sin(a \cos(y) + b) + c \] in \( x \),

\[ \sin(a \cos(y) + b) + c \] in \( y \) direction,

- shown in dashed outline (dark gray)

- Each drawn frame forms a field affects the component points within its field of the Gyroid surface, whose outlines are drawn in blue
We do not only find the structural order in *Object-oriented Programming* language corresponding to the one in geometries, but by extending this way of thinking, we also find that it integrates production and construction methods in detail into design at early phase of planning.

A building serves to provide environments for human beings to conduct activities, for which the requested dimensions are needed. If we could only see those needed dimensions and forget the usual names of their hosts, the form we need could be aggregated by various instances of one geometrical class. A parameterised constructing method of an ancient Pyramid.

Thoughts of this section are inspired from the following books:

*“Schloss - Palast der Republik - Humboldt-Forum”*  
- by Bruno Flierl, 2009

Keywords:  
*Berliner Schloss, Palast der Republik, Movement, CNC, Mass Production*
there have been two historical building standing here. First built in 1451, it was called “Schloss der Brandenburgischen Kurfürsten”63. This building, having been many times extended and later known as “Berliner Schloss”, had its glorious eras and shadowed moments through many historical events. Its existence stopped in 1950, after the government of German Democratic Republic decided to tear it down, partly due to its half destroyed building condition, and partly because of its once notorious political symbolism in the past. That was first time since its founding, between 1950 to 1973, the city Berlin, the historical capital of German volk, had nothing standing in its heart spot, Berlin Mitte. Nothing there until the East Germany started to building the next political symbolic building on the very location - the “Palast der Republik” (Palace of the Republic).

After the demolition, luckily “Berliner Schloss” still has a part of it remain standing around, however, this part has been moved from its original post to the place where it is hosted on another building which is, standing just fifty metres away, former named “Staatsratsgebäude”, and now the “European School of Management and Technology”. This special part, called Portal IV, had a honarable story for having avoided from being torn down.64

On the 9th of November, 1918, the time that we remember as the famous Novemberrevolution (November Revolution) in Germany, which ended the First World War and the German Empire, and established the Weimar Republik, and the day that Karl Liebknecht stood at the Portal IV of the Berliner Schloss, and announced the “sozialistische deutsche Republik”, two hours before the Republik was announced in the Reichstaggebäude by Philipp Scheidemann65. This historical moment saved the Portal IV from the demolished fate as the rest parts of the Schloss.

An appropriate symbolic building were searched for more then two decades, until the establishment of the Palast der Republik in 1976. Before the collapse of the German empire, entering this site for the commons was impossible, since it was the residence of the royalty. However, the Palast der Republik has opened its arms towards to the civilians. It was a building where not only the important political meetings were held but the rock concerts were also performed here, and so did those dancing classes, ice skating activities as well66. It was a building full of more cultural events, less politics, and for the first time the people of the city could really reach the heart of the city, until its ending phase starting in 200667.

Short before the German reunification, it is discovered that this building were contaminated with asbestos and it was shut down to the public on 19th of September, 199068. Although the asbestos had been removed and due to the procedure the building was stripped down to only its skeleton left. For a proper future on the site, there were numerous competition entries handed in. After a reconstruction of the Berliner Schloss was decided, it was either suggested in many entries69 or in artworks to keep the Palast der Republik building and to overlap and combine with the reconstruction partially - an artwork was even built with its facades printed on one-to-one scale canvases hung on the scaffoldings on the exact site. However, despite all these efforts, it still could not get the Palast out of its fate to be completely demolished. Again, the heart of Berlin has become void, and this time the emptiness lasts until 2012, the start of the construction of the won competition by Italian architect, Franco Stella, in 2009.

Between 2003, after the harmful asbestos had been removed, until the start of demolishment, Palast der Republik had, probably, lived its most glorious years. After the renification, along with the ‘keep it or not’ issue, more events, concerts or demonstrations and art works have been more than ever intensively happening in this building. A public building served with its most until it last day. Among all the memoribles, it, probably most being seen, is the picture that the Palace with a huge word instalation “ZWEIFEL” standing on the roof. An artwork, created by Lars Ramberg in 2005, meaning “doubt”, “perform[ed] new debates on national identity as well as the destiny of the Palace.”

After the announcement of the winning project for the competition of the re-

63 Schloss - Palast der Republik - Humboldt-Forum, p. 2
64 Schloss - Palast der Republik - Humboldt-Forum, p. 10
65 Schloss - Palast der Republik - Humboldt-Forum, p. 21
66 http://en.wikipedia.org/wiki/Palace_of_the_Repub-
67 Schloss - Palast der Republik - Humboldt-Forum, p. 15
68 http://www.larsramberg.de/1/viewentry/3890
MOVEMENT

shows momentarily how a person conducts his or her activity, and at this moment this movement relates the human body interactively either with one object or between objects. These objects serve this person in a way, spatially, that they provide the certain dimensions, with which the user can achieve his or hers activity.

For example, when we sit to dine, we need a height around 45 - 55 centimetres for holding our bodies in a resting position, “being seated”, in order to digest properly and also, in a social way, to conduct a talk, to communicate with the other persons over the dining time. For a better performance in this case, another object with certain dimensions in height and width is needed, in order to lay down our dishes to a position, say around 75 - 85 centimetres high and 70 centimetres wide, where we can easily reach in further conducting dining and communication. We usually call it a dining table. We can also easily find other needing dimensions elsewhere, such as having a height around 50 centimetres but with a relative wide in one dimension, around 120 centimetres for one person or 180 - 200 for a couple, and 200 centimetres long in another, where we put out whole bodies in resting situation, and we give it a sweet name for this moment: bed time.

However, our bodies are flexible, and we do not only have these rigid dimension-al choices. We can sit lower and lower, before touching the ground, we usually call it “squat” or “crouch”. Before we lie down flat, we can lean on some pillows, to keep our body in a inbetween position, where we usually read a bed story, or have some sun tan on the beach chair. Also, to the same 85 cm height, we can lean our hips on as well, to make an easying position for a not totally standing body, but half seated. It makes a so-called table neither for dining nor studying but kind of for resting, however not so relaxing like a chair. Even architectural elements, which are originally meant for other purpose than being used as a furniture, can be interpreted by some unintentionally usage. We find ourselves often sitting on the stair steps, waiting. We chat to each other on the balcony, maybe light a cigarette, and hold a beer with leaning selves to the handrails, loosening up. We lean forwards to a window, expecting. We actually put our bodies, literally, into fitting positions as much as we can to interact with our surroundings.

These un-named activities have probably defined our surroundings unintentionally through our creative interactions. However we usually only see those certain objects with certain names: chairs, tables, desks, sofas... etc., or in architectural way: doors, windows, columns, stairs... and so on. Other than these, we find no names.

Then, why should we only have certain activities related to certain object names? We actually can see and we actually conduct a “permeation” between movements to movements and from activities to activities.

Numerical Control, “is the automation of machine tools that are operated by precisely programmed commands encoded on a storage medium”, and nowaday, it is mostly controlled by computers as “CNC” - Computer Numerical Control. The precision and automation of CNC benefit a designer for a mass production from one prototype, but with numerous variations in size.

CNC production is not a new method, we see it in everyday life, especially in industry or product design. Utilising massive reproduction of material is also not new in architecture. We see its results at every inch of constructions. Ancient Egypte pyramids, Greek columns, Chinese wooden temples or a Catherine window in church, a simple brick wall at the front gate of a house, with standardised windows, doors, and repeating patterns of the roof bricks, the facade mosaic and the tiling in the bathrooms. However, thinking how to use efficiently the benefit, that CNC brings us due to the easier access to computer skills, in the architectural design process, leads us to a new era.

Once a Chinese writer, 徐志摩(Xu Zhimo), used the expression, “數大即是美”, to describe the beauty of a pattern, which is composed by a massive amount of repeating elements. He was expressing that his newly discovered feeling as seeing a flock of swallows flying across the sky in an ordered pattern. The aesthetics in patterns is now becoming more obvious and comprehensible in architectural designs.

Combining algorithmic modelling with design method, we transform the whole form into pattern regulations. By thinking small to form the whole, we start to focus on the compositions of one single element in detail and of the whole structural rigidity as well. Static is just one part of it. More in control would be also possible in applying materials for each element separately, and due to the similarity in geometry, a simplified and unitised build-up system can be applied on each element until to form the whole. This improves the efficiency in producing progress and makes the comprehension of a mass distinction imaginable even at the beginning phase of design.

The new way of form construction corresponds also to the philosophy of Object-oriented Programming. For each of the repeating composing elements, they make the same geometry class. Each instance, initialised from this class, has its own properties, of sizes and of its location information as well, and even its detail as a sub-class, which represents another geometrical prototype and many of which compose the whole form of one element. A system builds from the bottom up. Each of the detail instance has the information of its location relating to the others within an element, and each element has its location in the same manner to each other but under the whole form level. Each of them has a purpose, if one of them went missing, the form would not be complete, the structure cannot be stable and the goal of providing the certain dimension for user to achieve their purposes will not be satisfied. Then the meaning of the architecture, is also gone lost.
The complete point grids represent two surfaces of a spatial truss system. Each of the surfaces can be created by using "lofting" to connect between two end curves. By giving essential information, a wished curve form can be drawn by the "interpolating" points and a given curve degree - a method similar to calculating the "polyfit" function in NumPy.

The three-level structure of scripting - point grids, four vertices of a tetrahedron and three vertices of a triangle component of a tetrahedron facet - can not only make the complex algorithm clearer, but also lightens the load of computer calculation, and make each of the sub-structural script in other contexts reusable.

Firstly, at the upper level, a grid of points contains the information of all the three coordinates of each point. Each single coordinate is a real number, if we export them orderly, it will make just as light as a simple "csv" file. This make an end of one script. Importing the output csv file into the second script at the beginning, the workflow can continue on the information of the grid of points from the upper level, but gets rid of the resource consumption of the rerun of the whole grid structure.

In the second scripting level, the information of point grid can be partially read from the csv file and be dealt in one workflow, then be output to another csv file, since by each of lower level, the geometries will be detailed and the detailisation consists of more information, which burdens the computation. For example, in the second script, the goal of the outputs would be the grid points grouped in four for each tetrahedron, which multiply the points information more than doubled - a simple example would be to construct a cube form with 8 vertices, 6 tetrahedrons would be needed, i.e. an information of listed 24 points, which is two times longer than the input points.

To the next level, each of the four-point group indicates four sub-groups with the three vertices of one triangular facet. It is detailing. For complete the truss structure, the edges of a facet must be intact, but the inner part can be holed in different pattern, as long as the perforation keeps a certain distance away from the edges in order to form the thickness between two truss hinges. There will be more information added from the details.

It seems we are burdening ourselves with so much informations. However, all the information is manageable, and all the processes are reusable. All the points, tetrahedrons and the details are ordered in arrays, where each object has a numerised position or a name in the dictionary storage for indexing. A partial point grid can be retrieved, a certain tetrahedron can be picked out for changing its transparency on its facets and the pattern of perforation can be altered with targeting on one of the detailed triangles.

In an overview, a well-structured script can actually shorten the duration of production. Dealing with "how to realise the drawings" phase is actually moved forwards and integrated into design phase and progress.

Twenty years ago, to draw one tetrahedron on the computer screen was a breakthrough. It might still be limited to calculate the detailed tetrahedron truss system all at once, but in twenty years, we may not have to wait even so long, more detailed information of truss constructions can be processed easily and created in even smoother geometry.
In the main modelling environment Rhinoceros 5, with visualized programming plug-in Grasshopper and its add-on GhPython, Python scripts are written within buttons, in order to create user-defined functions, and to build conveniently workflows by collaborating with built-in functions.

Within the customised function buttons, libraries such as RhinoCommon, NumPy or randomisation are imported. The input information from previous workflows, such as point coordinates, curves, breps, ... etc., are output to the next step after being progressed within the script.

Keywords:
Rhinoceros 5, Grasshopper, GhPython, Python, NumPy, RhinoCommon
**POINT GRID**
For the computer numerical production control, the listed information, the output of each calculation embedded within scripts, can keep the traces of each element, even when the elements are accumulatively modified. Especially when the amount of elements is enormous and a specific item has to be retrieved for a special editing, an organised system would be quite helpful, which shows its significance especially at the final assembly procedure.

Different scales of the form prototype - tetrahedron - establish our goal forms in various detailed resolutions. The more complicated a form is, the smaller the basic size the prototype must be and so much does the amount of the unit has to grow. However, the tessellation is not just to rigidly divide, but the awkwardness in abrupt curvatures is to avoid by permeation in sizes.

Keywords:
CNC Production, Rhinoceros 5, Grasshopper, GhPython, Python, NumPy, RhinoCommon
SINGLE

Number mark: 301-3
Row position
Column position
Number of tetrahedron vertex
Number of triangle vertex
Welding edges
CNC cutting

DOUBLE

Connection
Connect to the adjacent tetrahedron
FIELD

of tetrahedra consists of the geometrical representatives, where the three triangle metal plates and glass slabs made by CNC-cutting in different sizes, according to the stored information in the algorithm. Each of the tetrahedra has its position information related to the whole field as well, which is stored within the properties of each of the class instances.

TRUSS FILED

is created between two edge curve pairs as illustrated. Firstly the maximal distance between top and bottom curves on each side of edge curves will determine where the most density of tetrahedron should fall on, in order to create a smooth at higher curvature. Secondly, a slight influence on each component point of the inbetween curves is coming from the higher maximum curvature difference, comparing upper and lower edge curve pairs.
TRUSS VARIATIONS

are shown as the following illustrations, which are varied in their initial unit length, therefore are different details however in the same whole surface form. To notice is, these variations are combined by two truss fields, between edge curve I over conversion curve then to the edge curve II. On the conversion curve, since the curve is the same one for both of the upper and lower truss fields, the divisions, either towards edge curve I or to curve II will be the same, thus the two fields are seamlessly bound to each other in order to form a stable truss system.
Two features of Palast der Republik are most memorable for us: one functional, the hexagonal auditorium with mechanism to be lifted up and down for multiple purposes, and the golden modulised facade, as we remember in the pictures reflecting an afternoon sun.

Two continuous surfaces represent the permeation from floor slabs to facade segments, and between sitting usage to structural supporting.

Numerous tetrahedrons bound with each other perform a permeation between different dimensions of the whole form and graduation in size of one single tetrahedron to another.

Keywords:
Rhinoceros 5, Grasshopper, GhPython, Python, NumPy, RhinoCommon
has been devoted by the winning competition entry by Italian architect Franco Stella. The planned spatial functions for the ground floor can be easily read from the schemes of the furniture arrangements on the floor plan.
CORRIDOR
OVERVIEW

1. Corridor
2. Facade
3. Ground Floor
4. Audience/Stairs
5. Upper Floor
Zeitportal

Floor

Perspective East

Perspective West

Top View

E

W

210 cm

85 cm

50 cm