TOPO-LOGY OF COM-PUTATION

CONRAD WEISE
INTERMEDIATE II

BACHELOR INTEGRATED DESIGN

AREA OF EXPERTISE DTF
UNDER THE SUPERVISION OF
PROF. DR. CAROLIN HÖFLER

KÖLN INTERNATIONAL SCHOOL OF DESIGN
FACULTY OF CULTURAL STUDIES

CONTENTS

CHA-OTIC STO-RAGE

ARTICLE 01

Amazon warehouses are a commonly occurring architectural sight in western societies. Just like other industrial buildings they arise from industrialized areas or in the outskirts of the next bigger city. To the outside the buildings are unreadable grey-shaded containers, mostly exceeding their surroundings by a great extend, ranging up to sizes of 100,000 square meters.

What hides behind the "Place your order" button are Amazon's self proclaimed fulfillment centers. An Amazon fulfillment center is the place where Amazon prepares its orders: Incoming items from third-party suppliers are stored and orders from amazon.com are shipped out. Items that are not directly available at the local fulfillment center are requested at a nearby warehouse; then gathered and shipped. Amazon uses its vast network of warehouses, so "Amazon prime; same-day delivery" orders can be completed within several hours. Amazon has over 200 warehouses in the United States alone. While Amazon‘s main business lies in e-commerce, it is expanding more and more into the direction of industry 4.0. It is heavily investing in the automation of their processes and researching on new mechanized ways to never stop fulfilling at its fulfillment centers.

While the Amazon business model itself is not hugely profitable, the part of Amazon that keeps it alive, is its mere scale and efficiency of operating. To keep up with its tight margins and its level of productivity, Amazon came up with its own

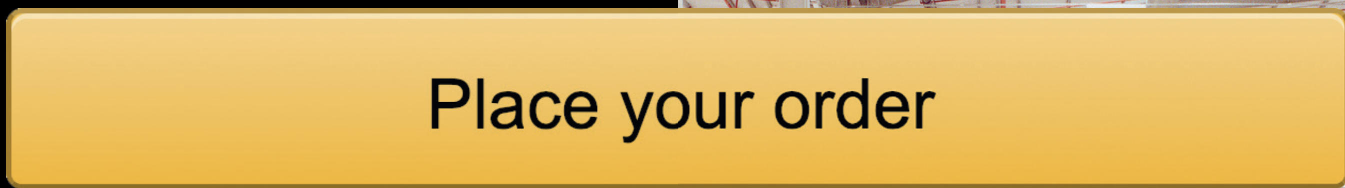

Figure 1.1: Place your order button

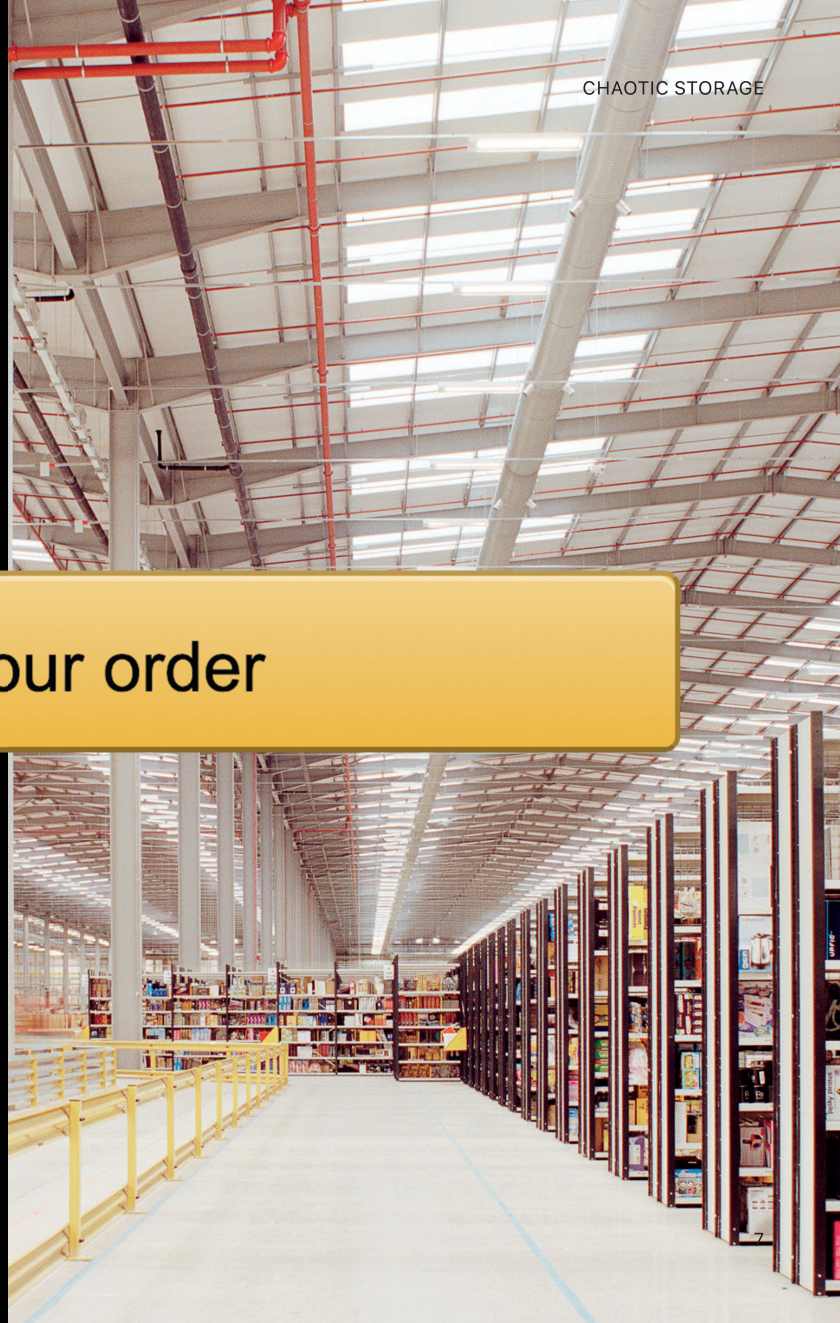

Figure 1.2: Amazon fulfillment center in Rugeley

structure to store items in its warehouses (Baraniuk, 2015).

Having a closer look at the depository at the fulfillment center in Rugeley, a small town in the middle of England, just the vast scale of it is intimidating: hundreds of hundreds shelves are packed closely to form aisles of uncountable items. It is arranged as an endless assemblage of consumer goods representing the supermarket of the 21st century. At the beginning of the aisles are light-yellow colored marks on the ground; this is where the human comes into play. The marks show where bins have to be placed after the picking ceremony is finished.

A "picker" is the most common job at a warehouse, namely a person that has to gather the specific items for an incoming order, collect them in a bin and pass it over to the packing station. The process of picking is a very interesting one. Unlike a regular storage system, items are structured by a method Amazon calls "chaotic storage" (Bridle, 2018). Its feature is that items are not stored logically by similarity or categories; but are, what it seems like, randomly placed. No items that could be confused are in the same shelf. This system is highly complex and is interily organized by the underlying storage management software. The structure is called chaotic storage, because it seems chaotic for the human eye, but from the algorithms point of view everything is in perfect order. Rather the algorithm would be quite unhappy with the inefficiency of human categories like size or aesthetic similarities. The algorithm is trained to minimize strolling and to maximize operation. So everytime an order is placed online and is received by the system a vast collection of Amazon's algorithms is being executed to predict, based on uncountable variables, the most efficient way to process the order. The most fitting fulfillment center is chosen and the handheld scanners of the local pickers start lighting up. All picker employees receive a handheld device at the start of their shift. This device is the interface, in which the algorithm can communicate with the chosen picker. The algorithm calculates the fastest route to collect all items for the specific order and sends instructions to the human-bonded interface. The picker then starts navigating the aisles with the only goal to complete the pre-augmented path of the algorithm. Everytime an item is picked, the item itself and the according bin is scanned, so that the software knows exactly its position and the state of the order.

With a phenomenon like the chaotic storage system it becomes apparent that computational systems are increasingly interfering within our physical spaces. Computation is not only present within these spaces but it is actively influencing them. Computational structuring will emerge ever more within the socio-spatial environments of today. To further understand the negotiation of space between humans and algorithms, it is important to first look at computation itself.

Figure 1.3: Amazon fulfillment center in Rugeley

SOFTWARE'S EVER GROWING (INVISIBLE) IMPLICATIONS

ARTICLE 02

Computation is all around us. Whether it is the self-reinforcing machine learning algorithm used by the local police to predict a next crime, or whether it exists in the form of algorithmically created YouTube videos for children to maximize view counts or whether it is the next hype technology that consumes more electricity than entire countries. In today‘s information age, software not only becomes an essential part of everyday life, but it determines most of it. Nevertheless software's implications are not always that visible or directly measurable. It seems like software is losing its direct materiality.

Software itself is written in various programming languages, ranging from high-level languages that are close to English, to low-level languages like Assembly, which is also called symbolic machine code, because it is so near to writing actual 1s and 0s. Software essentially always compiles down to binary code to be able to execute on its underlying hardware. Today's hardware architecture is based on one crucial component called the transistor. It basically acts like a switch that can be turned on and off, or that sets the voltage to 5 or 0 Volts. This component made it possible to write basic logic in so called logic gates. A constellation of multiple logic gates is called an integrated circuit. It is build to perform specific kind of logic operations and from there on the complexity of logic is as good as infinite.

So let me rephrase: software is losing its direct materiality, because since the beginning of computation it's physicality

Figure 2.1: First fully assembled transistor,
50mm in height

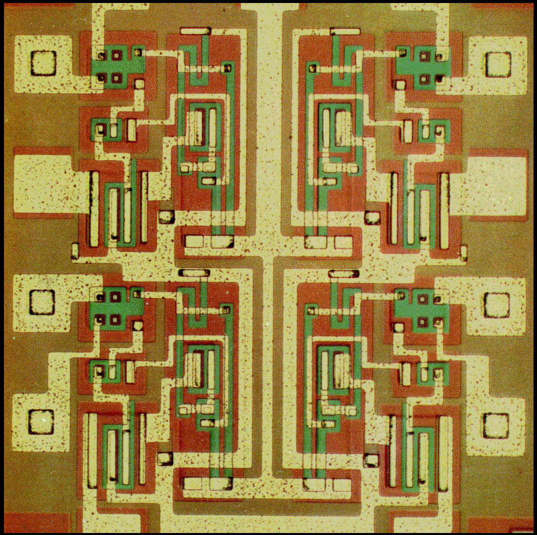

Figure 2.2: Integrated circuit,
multiple 7 nm sized transistors

has become increasingly smaller. In 1965 Gordon Moore, the founder of Intel, famously stated that the number of transistors in an integrated circuit will roughly double every two years. Since that statement the integrated circuit industry didn't slow down. Intel became so powerful that it drove almost every innovation in the field, leaving no space for competition while trying to catch up with its own expectations. The first transistor, made by Bell Labs, was roughly 50mm in height; nowadays transistor sizes down to 10 nanometers are the industrial standard (Beatrice Co, 2019). Physical computation became almost transparent: most of the computation is happening behind the user interface, embedded in integrated circuits consisting out of roughly 3 billion transistors, fiber-optic cables lay deep in the pacific, serving the invisible backbone of the internet, and almost everything is stored in the cloud, also known as the next data center nearby.

To further investigate the materiality of software and how it is able to blend in almost every environment, two properties of software appear important to examine more closely: its opacity and its adaptability.

I see the property of opacity closely related to the terms of transparency and opaqueness. Software has the ability to change the value of its opacity, hence becoming more transparent or taking on a more dense opaqueness. Software can only take on this transformation because it possesses a high

amount of adaptability. These two properties are directly linked to each other. Software is always in a state of hybridity encapsulating these two properties. Having in mind software's most abstract form of binary code and its hardware counterpart consisting out of logic gates, software is free from any context. At the level of compilation it is so reduced, that it only exists in two states: on and off. At this state it doesn't know anymore where it was defined or anything about its development culture. Software appears to exist in a non-place without any materiality (Trogemann, 2010). Hence software can choose its physical shape and where it wants to execute. Because of software's fully embraced rationality manifested through its internal logical physicality, it is able to survive in highly heterogeneous computational devices. Compiled machine code is a highly adaptable medium, which can cultivate in numerous environments.

To take a closer look at the famous metaphor of the cloud, the idea of software's opacity becomes more clear. The cloud or in general the concept of the internet is only possible because of the infrastructure of networked computational devices. In general the cloud or the website that is pinged is nothing more than just another computer. It is the highly complex infrastructure of these computers that makes the cloud seem so persuasive and believable. The data centers, which are gatherings of computational devices, are just like the Amazon

Figure 2.3: Eastern Telegraph Company network in 1901

warehouse: massive grey boxes that are not perceivable from the outside. They are build in the industrial areas of cities and are highly secured, making it a no-go zone for everybody. These data centers are interconnected through fiber-optic cables ranging up to the diameter of the earth, based on an infrastructure which exists since the early 1900s (Atlantic Cable , 2012). The fiber-optic technique is using light to transfer computational data with the speed of light. Roughly 500 trillion bits computational data; the approximate mass of data traveling the internet per second. The infrastructure of the internet enabled computation to change its presence to be in the best rational conditions, most probably far away from the actual affected human being.

On the side of its e-commerce business, Amazon also plays a big role in the cloud computing industry. With Amazon Web Services, Amazon serves about 30% of the whole internet traffic. Cloud computing essentially consists of giving access to data centers and its computational devices serving various types of services. Comparing two images of an Amazon Web Services data center and an Amazon warehouse, it is not distinguishable which one is which: On one side the computation is being executed and on the other side the computation is manifesting itself with its implications in our physical world.

It seems like software's persuasiveness only becomes unsure when it can not adapt to the environment, hence when the in-

terplay of its opacity and adaptability is collapsing. Whether it be the next keystroke that is not showing up on the screen, because Google Docs lost the connection to its beloved data center in Frankfurt, or whether it is the bus ticket machine at a local airport near Venice, showing the Windows Vista taskbar at the bottom of the screen, revealing the underlying operating system. Only in these moments Heidegger's "readiness-to-hand" (Zuhandenheit) is broken and the software becomes present-at-hand (Vorhandenheit). The philosopher Martin Heidegger states that a person who is using a hammer, and is in the act of hammering, is actually not perceiving the hammer. For the person using it, the hammer is not identified as an object; the hammer doesn't exist as such. Rather it exists in the unconscious readiness-to-hand background. According to Heidegger the hammer itself becomes only perceivable as an actual object, when there is some kind of breaking down or unreadiness-to-hand occurring (Winograd and Flores, 1987). For me, the ticket machine showing its underlying operating system is the same as a hammer losing its head. In both situations the object that is used in the moment, does only exist as an entity when it breaks. Only then the object shows it's hidden properties so that we can reflect on them. While software doesn't break, it is not identified as such; it merely doesn't exist.

A similar approach is seen from the philosopher Bruno Latour. In his work "Pandora's Hope", hes described the term blackboxing as:

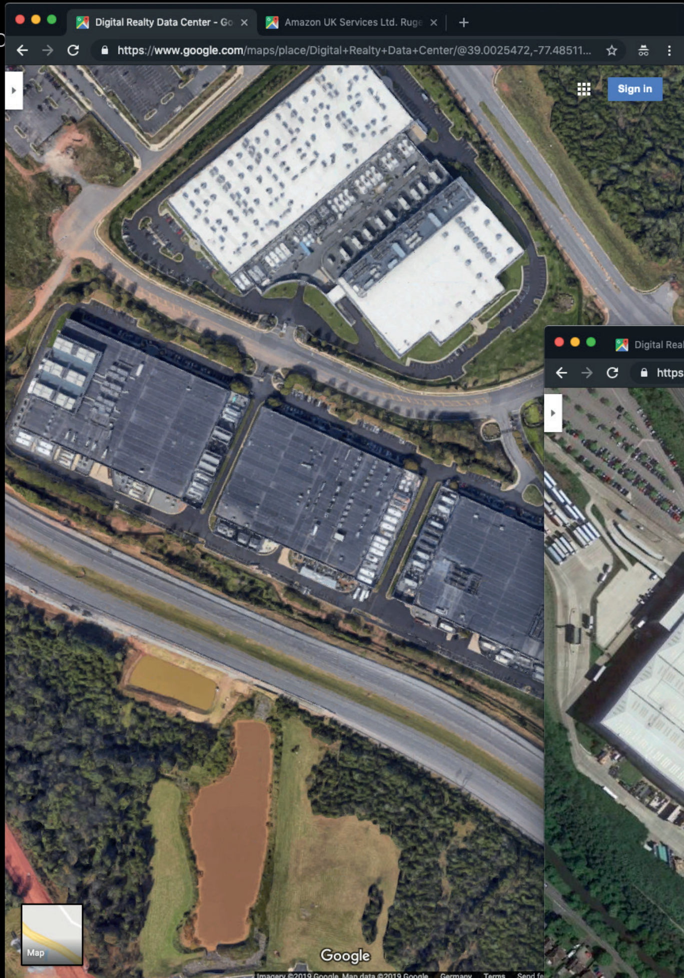

Figure 2.4: @39.0026533, -77.487622

Figure 2.5: @52.7532745, -1.9203118

> An expression from the sociology of science that refers to the way scientific and technical work is made invisible by its own success. When a machine runs efficiently, then a matter of fact is settled, one need focus only on its inputs and outputs and not on its internal complexity. Thus, paradoxically, the more science and technology succeed, the more opaque and obscure they become. (Latour, 1999, p. 304)

So in support of Heidegger's concept of breaking down, software doesn't only show itself when it loses, but on the contrary, the more it wins the more it moves in the background never revealing itself. With software's ability to adapt to various environments, reinforcing itself success, exposing the dangerous implications of computation will become increasingly difficult.

SOFTWARE AND HUMANS CONSTRUCTING SPACE TOGETHER

ARTICLE 03

On the contrary to software's ever more vanishing materiality, its invisible implications have risen a great deal. With that the necessity of software's localization is more important than ever. I see the localization of software as an analysis of where software's imprint, physicality and implications show themselves in space, while not disregarding its rational algorithmic origin. To successfully locate software in space, it is important to build upon the ontology of space defined after the so-called spatial turn.

The spatial turn was an intellectual phenomenon that occurred in the social sciences and humanities, fundamentally challenging the definition of space. In the 1960s space was regarded as being absolute; measured within its three dimensions x, y and z. Space was essentially reduced to its locational geometry; regarded as being devideable into squares and grids. It was conceived as a container. Humans and their actions were considered within the theories, but they were merely placed into space. In the absolute ontology it was regarded that humans didn't affect space directly. But this changed from the 1970s onwards with the introduction of concepts around the relativity of space. Within this approach of relational space, the main opposition was that the scientific absolute space was constrained from social meaning and political purpose. Relational space distanced itself from the neutrality, passiveness and unbiased way of looking at space. On the contrary they proposed to look at space as being an active network of social relations which is in constant change. For them

> space was not an absolute geometric container in which social and economic life took place; rather, it was constitutive of such relations. (Kitchin and Dodge, 2011, p. 67)

The two authors, Rob Kitchin and Martin Dodge, of the book "Code/Space" go a step further by looking at the non-representational theory of the geographer Nigel Thrift.

Thrift argues that spatial practices are often performed without cognitive and rational thought and that these human practices are complemented by other actants. Whether it be animals, objects, ticket machines, transistors or the fiber-cable networks in the pacific. In particular, Thrift is interested in the phenomenon when the actants are determined by software, and how then, space is created automatically, without the interference of humans.

Kitchin and Dodge themselves define their space ontology as following:

> [...] space is constantly brought into being as an incomplete solution to an ongoing relational problem. (Kitchin and Dodge, 2011, p. 71)

They argue their statement with the concept of transduction. Transduction being a process by which things transfer from

one state to another. Actually originating from biology, the term is used to describe the transfer of genetic material from one cell to another by viral infection. With the work of Adrian Mackenzie, Kitchin and Dodge explain that

> [...] through transduction, a domain structures itself as a partial, always incomplete solution to a relational problem. (Mackenzie, 2003, p. 10)

This phenomenon is what they call the transduction of space. Looking at space from this perspective, they argue, space could be described as a stream of occurring relational problems. They theorize space as being constantly in a state of becoming. To navigate space would be to provide a solution to the next appearing problem. This solution will always be incomplete, hence creating another relational problem; and with that a stream of actively constructed space. Like Thrift, their observation is that these problems can be solved by computation. They state:

> software is increasingly providing the solution to relational problems. (Kitchin and Dodge, 2011, p. 73)

With that, software has the ability to actively construct space; a so called code/space. A space that sees the existence of computation as inevitable. For a code/space software not only becomes a crucial part of space, but it becomes essential. It describes space that would not exist without software. The mere existence of a concept, such as code/space, reflects the impact software has on our everyday life. In regards to the phenomenon of the infrastructure of the internet, it appears that the entire world becomes a code/space.

FROM HU-MAN-CEN-TERED TO MACHI-NE-CENTE-RED SPACES

ARTICLE 04

With the concept of code/space in mind, looking back at the Amazon warehouse, it becomes apparent that the transduction of space through software is happening at every scale of the process. Software is an active actant within the structure of the space, creating relational but also physical space. Regarding software's origin, the solutions it can provide to create space through transduction are strictly rational and algorithmically limited, and thus is the space constructed by it. Looking at the architectural structures of the warehouse, software's rationality is reflecting onto every part of it.

From an interior bird's-eye view the warehouse looks like being divided into several squares and grids, strongly resembling the definition of absolute space before the spatial turn. Whether they are the processing stations, the aisles of shelves or the endless stretch of conveyor belts. Everything is ordered in reference to and by the grid's marks, all structured in 90 degree angles to each other, which are not only the light-yellow ones for the carts, but also blue and orange ones, which appear to mark transport and storage areas. The marks are just the visible layer on the ground, being only a small part of the underlying grid, so the human can understand and follow the strict rationality of the space. The warehouse is build to maximize its surface for storage, production and processing, leaving only narrow aisles for humans to navigate. There is not a single square in the grid without a direct function to fulfill the fulfillment center's will. Having a closer look at the broad overview of the warehouse, patterns and repetition are a key characteristic of the space. Everything is laid out for repetition; repetition being one of the strongest qualities of computation. But it would be quite unfair to link the high rationality of the space to the soley existence of computation. Highly rational spaces existed way before computers could interfere with the creation of physical space.

In the architectural discourse, the rational structuring of space is known for quite some time now. With modern architecture and its values rationality became the most important asset of the architect. Modern architecture was established in the first half of the 20th century and was mostly grounded upon new materials and progress in construction techniques like reinforced concrete, steel frames, curtain walls and ribbon windows. It rejected its predecessor's ideas of ornamentation, architectural styles in general and focused on practising minimalism. Leading architects wanted to construct purely functional spaces and many endorsed the formula "form follows function", which implies the underlying rationality; namely to construct physical space based on reason (RIBA, 2019).

The architect and urban planner Charles-Édouard Jeanneret, also known as Le Corbusier, is considered to be one of the pioneers of modern architecture. His absolute devotion to mathematics and metrics is shown in his anthropometric scale called the "modulor". Closely related to the Fibonacci numbers, the golden ratio and Leonardo da Vinci's "Vitruvian Man", Le Corbusier tried to map mathematical proportions onto the human body, more precisely a man's body, to reveal a new system, which would improve the structure and functionality in his architecture.

In 1925 Le Corbusier presented a plan, Plan Voisin, to rebuild the center of Paris at the International Exhibition of Modern Decorative and Industrial Arts. The replanning proposal showed a highly rationalistic vision of Paris. The center would resemble a large scale rectangular concrete surface functioning as an open space. With twenty-four cross shaped skyscrapers he would concentrate the business area vertically to be able to serve the large open space in the center. Le Corbusier always paid special attention to the industrialization happening at the same time. In his very similar structured plan of "Une Ville Contemporaine" (A Contemporary City), Le Corbusier would separate the industrial and the residence area with a park and would also include cross shaped skyscrapers in the center. As an architect of modern architecture, he would always consider the function of a city first; the function of a working and growing economy.

Le Corbusier was also a painter and writer, actively taking part in the art and architecture discourse. With the cubist painter Amédée Ozenfant, he started writing about their new artistic movement "Purism", which is a variation of Cubism, and about his opinions on art and architecture in general. The articles, named "L'Esprit Nouveau" (The New Spirit), would essentially lead to his book "L'art décoratif d'aujourd'hui" (The Decorative Art of Today). The book would stand for his reluctance to decorative art or any kind of decoration, further resembling his stand in modern architecture and his rational and functional-driven take on the world (Le Corbusier, 1987). In the book he states his ideal of a worker's life:

> The ideal is to go work in the superb office of a modern factory, rectangular and well-lit, painted in white Ripolin; where healthy activity and laborious optimism reign. (Le Corbusier 1987, p. 75)

With the replanning of Paris and his special attention to the industrialization and its worker economy in mind, this statement of Le Corbusier is a quite astonishing accurate description of an Amazon warehouse environment of today.

Leading us to the industrialized spaces, and looking at these spaces directly, we find even more similarity to our case of an Amazon fulfillment center. With the second industrial revolution mass production and assembly-line architecture became very popular. The industrial architect Albert Kahn had a big influence on the new design structures of industrial factories in the U. S., especially in Detroit, Michigan. His works range from the Fisher Building, being a 30-story Art Deco style skyscraper, to several large scale automobile manufacturer factories, like the General Motors Building or the Ford Assembly Plant. He is known for introducing reinforced concrete walls and perfecting the assembly-line architecture.

For example, looking at his works of the Detroit Arsenal or the Ford River Rouge Complex, direct similarities to the Amazon warehouse become apparent. Architecturally the sights have very similar structures and the previous observations of the rational physical space of the Amazon warehouse are almost completely transferable onto Kahn's factories. From the outside the sights are rectangular industrial complexes, forming impenetrable physical cubes. Inside the similarities continue: marks on the ground, resembling the rational grid, narrow aisles for transportations, large scale areas for storage and production, high ceilings with steel beams forming the roof structure, which lets natural blend with artificial lighting and patterns, and repetition and patterns permeate the whole complex. The structures focus on functionality and production. New spaces were established; so-called machine-centered spaces. Spaces which were designed with the incentive to build and produce with or by machines; and humans were just the necessary by-product to fulfill the architecture's function.

Rational and functional-driven thought and the implementation of it in physical space is not a new phenomenon that appeared through computation. Since the early industrial times, humans created these machine-centered spaces without the use of software or any computational systems. The difference between the machine-centered spaces from today and the machine-centered spaces from Albert Kahn and co, is the connection to their relational space. Architecturally they seem very similar, but the interwoven relational space is where we can spot the difference and also new structures of the space surrounding an Amazon fulfillment center.

PHYSICAL SPACE BECOMES INCOMPREHENSIBLE FOR HUMANS

ARTICLE 05

To showcase the ever growing complexity of the interconnection of relational and physical space, a look at one of the first factories ever constructed gives great insight. In 1775-1778 the architect Claude-Nicolas Ledoux build the “Les Salines Royales” complex (The Royal Saltworks) near the large forest of Chaux in France (Fabrizi, 2016). Its function was to process the nearby wood and water to extract the, at the time very precious, salt deposits of the water. The complex consists of several housing groups arranged in a semicircular shape. The curve aligns the housing area for the workers, with the entrance to the complex in the middle, whereas the furnaces and processing factories were built along the straight side of the complex. In the middle of the straight side, thus the focus point of the complex, Ledoux placed “la maison du directeur” (The House of The Director).

This architectural structure is also called panopticon and it embodies strong ideological and political characteristics. First introduced in 1791, as a proposal of a new prison, the panopticon was designed as a system of control and to optimize surveillance. With the observer in the center, being able to observe all surrounding entities at once, the entities can never be sure if they are being watched or not. It leaves them in a constant state of being monitored and the observer in a constant state of control. The panopticon-shaped structure of the Royal Saltworks shows a clear working hierarchy. From the center of the complex, the director’s total control is being practised

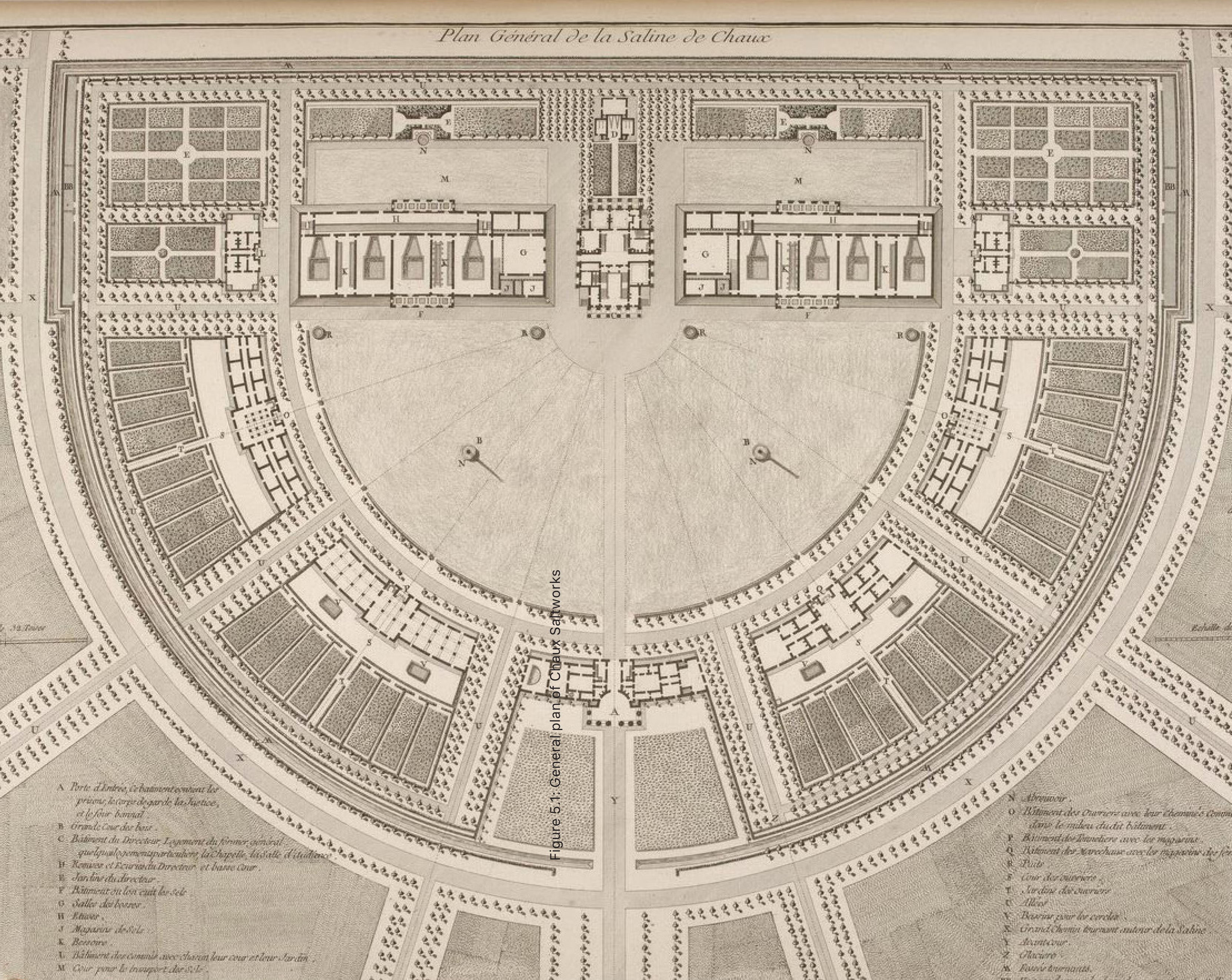

Figure 5.1: General plan of Chaux Saltworks

onto the surrounding worker houses (Foucault, 1995).

With spaces like the Royal Saltworks the interwoven social interconnections are constructed into the physical space. The power dynamics between director and worker are distinct and certainly readable from the outside. The physical space clearly resembles the interwoven relational space. At that time, the relational space and the physical space still aligned; they were understandable and made sense together. But through the growing effects of capitalism and with industrialization's decentralization of production processes, the directors of our complex systems are not that easy to pinpoint anymore. Processes and structures and their physical spaces became increasingly decentralized and with that also their relational spaces. Social interconnections and power dynamics became increasingly hard to understand. Already in the industrial works of Albert Kahn, a tendency in that direction is noticeable. Industrialized spaces became increasingly rational and pure to functionality, while on the other hand the relational spaces grew in complexity. The increasingly complex network of decision-making processes were leading to the relational and physical space drifting increasingly apart. A spatial dissonance occured.

Returning to the Amazon warehouse environment and taking computation back into the architectural equation, this spatial dissonance reached a climax. The Amazon fulfillment center

serves an increasingly reduced and rationalized physical space, whereas the relational code/space reveals utterly complex structures. Not differently, but even more profoundly is Amazon's network of data centers, which power and control the warehouses. The data centers are highly rationalized physical spaces, embodying the rational origin of computation.

Inside these structures code/space is camouflaged as a strictly absolute space, but it‘s true identity of the complex relational space it is, cannot be denied. The data centers function as the underlying network of decision-making processes, being a substitute for the director's house. But not just one; more precisely a vast decentralized version of it. The complex around the Amazon warehouse forms a decentralized panopticon, with computation being in control of the decision-making process. It actively replaces cognitive work and resembles the director of the 21st century affecting numerous people.

Within the domain of artificial intelligence, recently made progress in machine learning gave computation the power to formalize code by itself. Given a set of input and correlating output data, a computational system can, through the operation of artificial neural networks, recognize patterns in the data, to formalize rules by itself. This new trend of machine learning algorithms is already heavily implemented in the computational world of today, and thus also a major part of the environment surrounding the Amazon fulfillment centers.

While the new technology brings great improvements in the field of data science and analytics, it deeply contributes to the complexity of the code/space, hence the increasing segregation of physical and relational space. Through its independence and self-learning capabilities it performs as a major opaque actant within the code/space. Decisions made by a machine learning model are hardly comprehensible and almost impossible to trace back.

Within these models, the recognized patterns, hence the sub-decisions, which are combined to formulate a prediction, are stored in a so-called latent space. In this latent space, the model builds its own version of the world, categorizing it in important and less important parts. Computation builds its own space; made by it and only meant for it. The space is a multi-dimensional data space to represent and structure the neural network's formalizations. It is only hardly understandable for a human, and visualizations exceeding the three-dimensional euclidean space are hard to imagine. Nevertheless this space might be the only option we have to further understand our complex relational spaces.

In 1917 the management engineer Frank Gilbreth published "Applied Motion Study: A Collection of Papers on the Efficient Method to Industrial Preparedness" as an exploration of industrial working operations to investigate and optimize movements of workers in physical and temporal space. Equipped

with a light source and the technique of long-exposure photography, the works of his motion study were able to visualize before unseen movements in physical space and thus creating a visual connection between the relational and physical space. From the photographs Gilbreth would later build physical models, to be able to analyse the workers movements in three-dimensional space (Gilbreth, 1917).

The analysis made by Gilbreth are similar to the analysis of a neural network algorithm. Instead of tracking human movements, it would track software movements, hence its decisions. And with that it could be seen as a physical representation of the latent space. The motion studies could be used to navigate the latent space and thus maybe also parts of the complex relational space surrounding it.

Concepts like Amazon's chaotic storage system are first occurrences, where parts of the latent space leave the machine and enter our physical space. Computation is building its own version of physical space. It structures this space with principles and techniques a human cannot grasp. Its rationalistic and cryptic way of working is manifested in it. To say it in Latour's terms: the opaqueness of the black box is leaking into physical space.

For now the incomprehensible world of computation was reduced to the inner workings of integrated circuits or machines. But with spaces like the Amazon warehouse its rationality and opaqueness is mapped onto our physical space.

What will happen to our understanding of physical space, when it becomes incomprehensible and impossible for us to understand?

Atlantic Cable. (2012). Submarine Cable Route Maps. History of the Atlantic Cable & Undersea Communications. Retrieved from http://atlantic-cable.com//Maps/index.htm on March 25, 2019.

Baraniuk, C. (2015). How algorithms run Amazon's warehouses. BBC Future. Retrieved from http://www.bbc.com/future/story/20150818-how-algorithms-run-amazons-warehouses on March 25, 2019.

Beatrice Co. (2013). History of The Transistor. Bell Labs History. Retrieved from https://beatriceco.com/bti/porticus/bell/belllabs_transistor.html on March 25, 2019.

Bridle, J. (2018). New Dark Age: Technology and the End of the Future. London: Verso.

Fabrizi, M. (2016). The Ideal City of Chaux by Claude-Nicolas Ledoux. Socks Architecture. Retrieved from http://socks-studio.com/2016/11/09/the-ideal-city-of-chaux-by-claude-nicolas-ledoux-1773-1806/ on March 25, 2019.

Gilbreth, F. (1917). Applied Motion Study: A Collection of Papers on the Efficient Method to Industrial Preparedness. New York: Sturgis & Walton Company

Kitchin, R. and Dodge, M. (2011). Code/Space: Software and Everyday Life. Cambridge: The MIT Press.

Latour, B. (1999). Pandora's Hope: Essays on the Reality of Science Studies. Cambridge: Harvard University Press.

Le Corbusier. (1987). The Decorative Art of Today. Cambridge: The MIT Press.

Mackenzie, A. (2003). Transduction: Invention. Innovation and Collective Life. Lancaster: Institute for Cultural Research.

RIBA. (2019). Modernism. Explore Architecture. Retrieved from https://www.architecture.com/explore-architecture/modernism on March 25, 2019.

Trogemann, G. (2010). Code und Material: Exkursionen ins Undingliche. Vienna: SpringerWienNewYork.

Winograd, T. and Flores F. (1987). Understanding Computers and Cognition: A New Foundation for Design. Norwood: Ablex Corporation.

Figure 1.1: Amazon.com. Screenshot from checkout. Retrieved from https://www.amazon.com/ on March 25, 2019.

Figure 1.2: Ben Roberts. Amazon Unpacked. Retrieved from https://www.benrobertsphotography.com/work/amazon-unpacked/ on March 25, 2019.

Figure 1.3: Ben Roberts. Amazon Unpacked. Retrieved from https://www.benrobertsphotography.com/work/amazon-unpacked/ on March 25, 2019.

Figure 2.1: Beatrice Co. History of The Transistor. Retrieved from https://beatriceco.com/bti/porticus/bell/belllabs_transistor.html on March 25, 2019.

Figure 2.2: Dgarte. Wikipedia 5 nanometer. Retrieved from https://en.wikipedia.org/wiki/5_nanometer/ on March 25, 2019.

Figure 2.3: TeleGeography. Eastern Telegraph Company. Retrieved from http://atlantic-cable.com//Maps/index.htm on March 25, 2019.

Figure 2.4: Google Maps. @39.0026533, -77.487622. Retrieved from https://www.google.com/maps/ on March 25, 2019.

Figure 2.5: Google Maps. @52.7532745, -1.9203118. Retrieved

from https://www.google.com/maps/ on March 25, 2019.

Figure 5.1: Socks. The Ideal City of Chaux by Claude-Nicolas Ledoux. Retrieved from http://socks-studio.com/2016/11/09/the-ideal-city-of-chaux-by-claude-nicolas-ledoux-1773-1806/ on March 25, 2019.