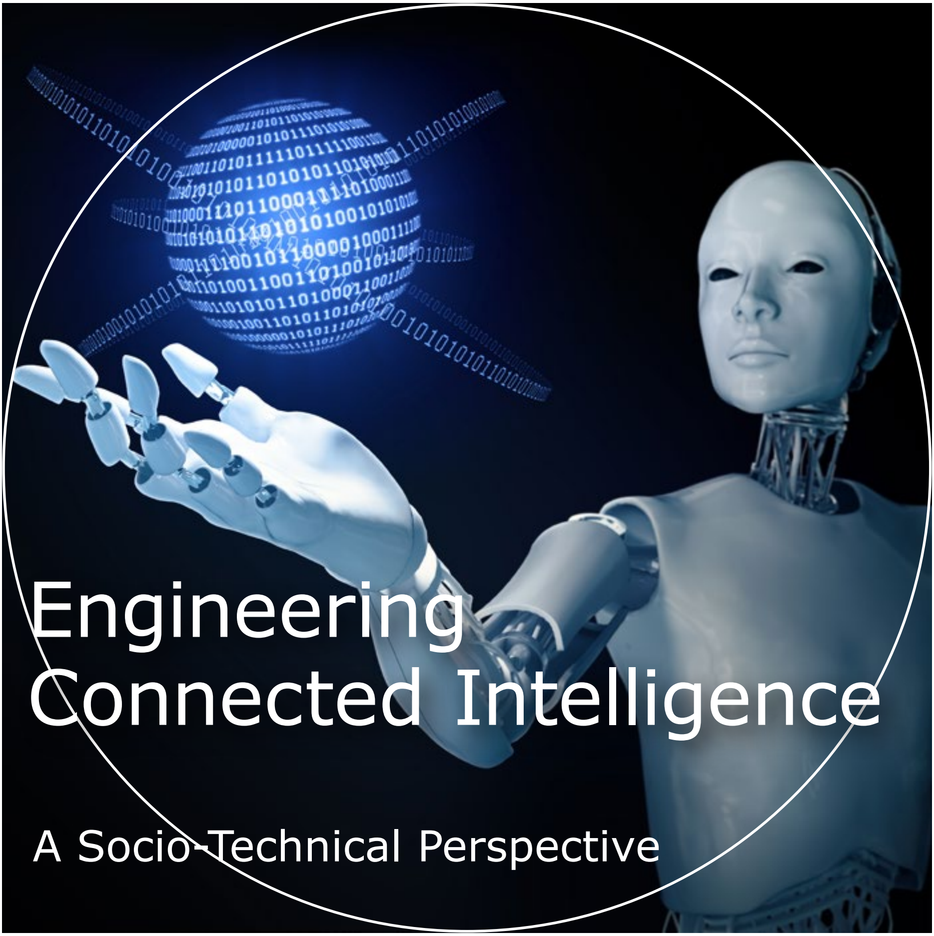

Engineering Connected Intelligence

A Socio-Technical Perspective

Prof.dr. Bedir Tekinerdogan

Inaugural lecture upon taking up the position of Professor of Information Technology at Wageningen University & Research on 2 February 2017

WAGENINGEN UNIVERSITY & RESEARCH

Engineering Connected Intelligence

A Socio-Technical Perspective

Prof.dr. Bedir Tekinerdogan

Inaugural lecture upon taking up the position of Professor of Information Technology at Wageningen University & Research on 2 February 2017

WAGENINGEN
UNIVERSITY & RESEARCH

DOI 10.18174/401115
ISBN 978-94-6343-049-4

Engineering Connected Intelligence

A Socio-Technical Perspective

Introduction

Esteemed Rector Magnificus, dear colleagues, students, family, and friends

We are guests in this world. We as the human species actually just recently arrived on this planet, since the existence of the universe about 15 billion years ago. Assuming that the Earth was created one year ago, the human species would then be only 10 minutes old, while the industrial era started two seconds ago [1]. It is in these last two seconds that we have made a drastic impact on the planet. In these last two seconds, we face with several grand societal challenges including health, demographic change and wellbeing, food security, sustainable agriculture and forestry, clean and efficient energy, resource efficiency and secure society. We have now begun even questioning whether the earth will survive the next few 'seconds'. On the other hand, technology has progressed exponentially and appears to provide an opportunity to cope with these challenges.

In this lecture, I will provide a socio-technical perspective on the advances in engineering and technology, the resulting industrial revolutions, the increased level of smart systems in the last decades, and the interconnection of these systems leading to a global connected intelligence.

From Craft Production to Mature Engineering

From its existence on, mankind has always had to face with different challenges on earth and has put much effort to survive in the history. In ancient times the basic needs of man were shelter, food gathering, agriculture, domestication of animals and hunting. To meet these needs, artefacts and devices were developed to solve practical problems and make natural resources more useful. Engineering, that is, the production of artifacts for practical purposes has been an early and continuing necessary profession of mankind. Obviously, in the early history, engineering was not that sophisticated as we experience it today. In the early societies the production of artifacts was basically done by hand and this is the reason why these societies were called *craft-based societies* [2][3]. In these contexts, there is no prior activity of describing the solution like drawing or modeling before the production of the artifact (Figure 1). Further, these early engineers had almost no knowledge of science, since there was no scientific knowledge established per today's understandings. Hence, the production of the artifacts was basically controlled by tradition, which was characterized by myth, legends, rituals and taboos and therefore no adequate reasons for many of the engineering decisions can be given. In fact, the available knowledge related with the craft process was stored in the artifact itself and in the minds of the craftsman, which transmitted this to successors during apprenticeship. Innovation was slow and largely missing and the form of a craft product gradually evolved only after a process of trial and error, heavily relying on the previous version of the product. The form of the artifact was only changed to correct errors or to meet new requirements, that is, if it is necessary. In a sense, there was thus little consciousness about the engineering activities, which is the reason why such engineering processes are termed as *unself-conscious process* [2].

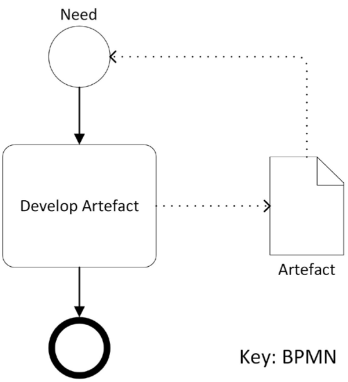

Figure 1. Primitive problem-solving process of mankind applied for ten thousands of years in history (Key: BPMN [4])

History shows that the engineering process evolved gradually and became necessarily *conscious* with the changing context. It is hard to pinpoint the exact historical periods but over time, the size and the complexity of the artifacts exceeded the cognitive capacity of a single craftsman and it became very hard if not impossible to produce an artifact by a single person. Moreover, when many craftsmen were involved in the production, communication about the production process and the final artifact became important. A reflection on this process required a fundamental change in engineering problem-solving. This initiated the necessity for drafting or designing whereby the artifact is represented through a drawing before the actual production (Figure 2 left). Through designing, engineers could communicate about the production of the artifact, evaluate the artifact before production and use the drafting or design as a guide for production. Currently, design plays an important role in all engineering disciplines.

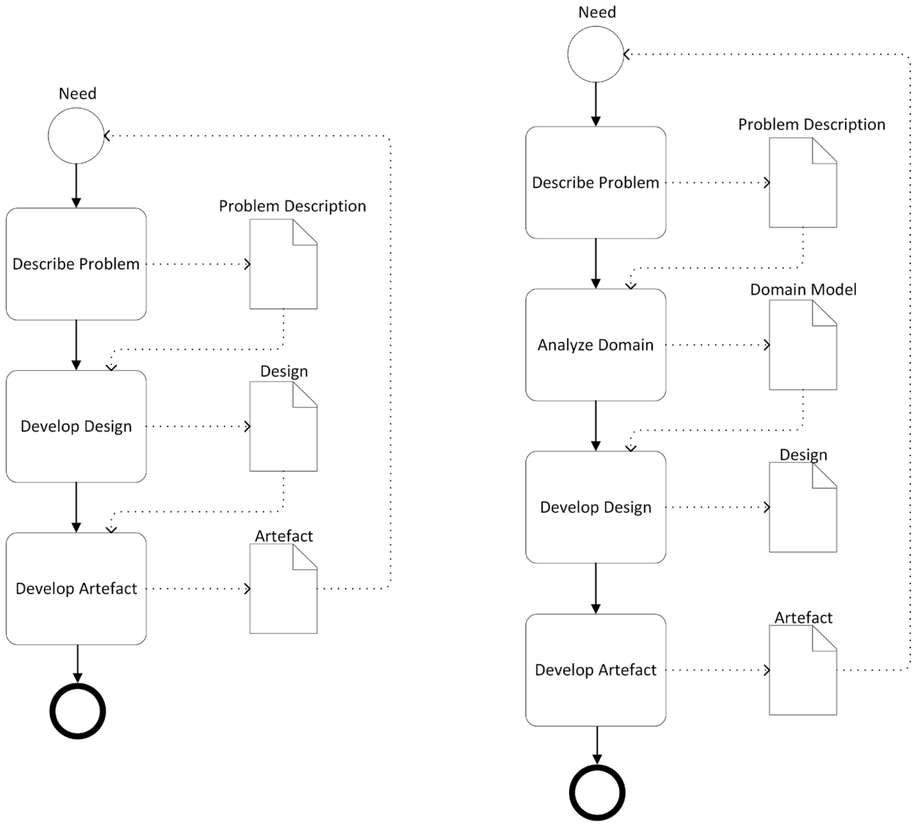

Figure 2. Design-driven problem-solving process (left)
Domain-driven problem-solving (right)

Design itself is not sufficient for engineering but usually additional scientific knowledge is required to develop the required artefacts (Figure 2 on the right). Scientific knowledge includes the body of empirical, theoretical, and practical knowledge that does not only serve to understand the natural world but also is important for guiding and optimizing the production of artefacts. In the early societies, scientific knowledge was lacking but over time it gradually evolved while forming the basis for the engineering disciplines. Currently, mature engineering is supported with scientific knowledge that has been compiled in several handbooks that guides the engineer in developing the artefacts [3]. With scientific knowledge, different alternative design solutions can be developed for solving the required needs [5]. Figure 3 shows the most mature engineering process that adopts a design-driven approach based on scientific knowledge, and targets the selection of feasible design alternatives with respect to functional and quality requirements [3].

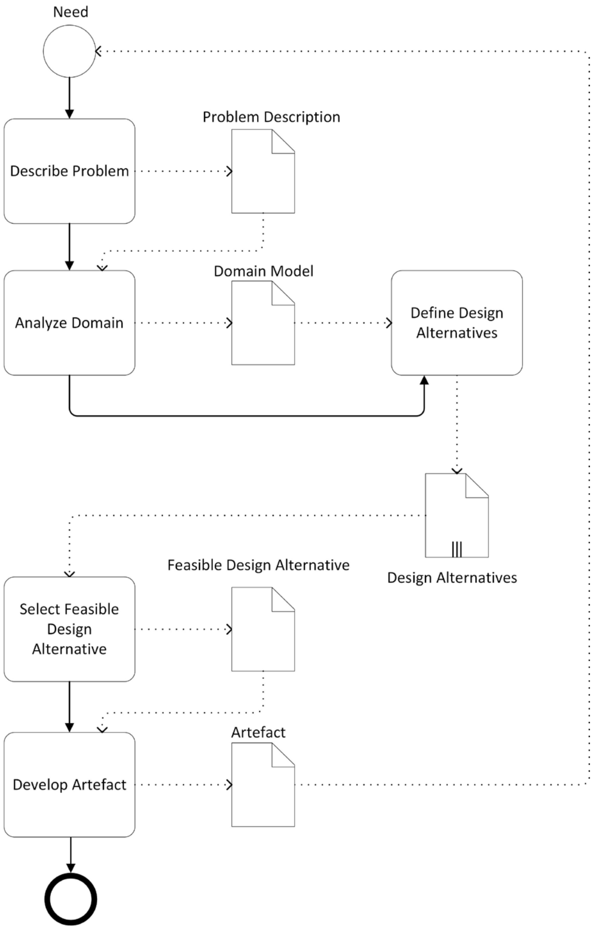

Figure 3. Model of Mature Engineering

In sum, we can state that engineering has evolved from primitive problem-solving to more advanced problem-solving. Further, we can observe that the more advanced the engineering approaches were, the higher was the observed impact on the society. In this context, the steam engine developed in 1769, was one of the results of advanced engineering that marks the initiation of the beginnings of the **First Industrial Revolution**, which on its turn implied the transition from a predominantly agrarian and rural society to an industrial and urban society [6] [7]. Prior to the Industrial Revolution, production was often done in individual's homes, carrying out laborious tasks using hand tools or basic machines. With the mechanization, the previously done laborious tasks were now located in the same factory and with the help of machines productivity could increase rapidly. This industrialization had a disruptive impact in different business domains and played an important role in the improvement of transportation and communication.

The First Industrial Revolution was soon followed by the **Second Industrial Revolution**, also known as the Technological Revolution, including further advancements in manufacturing and production technology [7]. The advent of electricity, the establishment of a machine tool industry, the development of methods for manufacturing interchangeable parts and the invention of mass production were important characteristics of this phase. As a result, products were produced faster and in a more efficient way, and the production process became increasingly routine and specialized.

The two industrial revolutions had an important role in the history of mankind since these helped to escape from the so-called *Malthusian catastrophe* [8]. In his essay, Thomas Malthus argued that human population is growing exponentially, while the earth's resources are growing at a much slower rate. Since the earth's resources would no longer be able to support such a large population, he predicted that in the near future the world had to face with long periods of famine, and mankind would be forced to return to subsistence-level conditions (Figure 4). The only way to overcome this problem was to use preventative methods and control the population growth. Factors such as natural disasters, famine and wars would bring the population down, but this would only be temporary and the problem as such would remain.

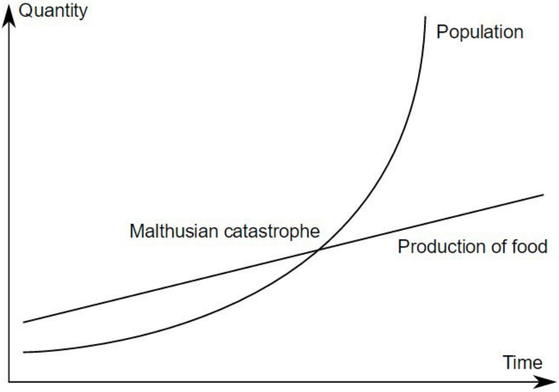

Figure 4. Predicted Malthusian Catastrophe which was avoided by First Industrial Revolution

Malthus' theory was right in several aspects but it did have its limitations. Malthus looked primarily at two main factors: population growth and agricultural production. Malthus did not, though, consider the unforeseen advances in technology that were triggered by the industrial revolution [6]. Yet, the development of new and innovative technologies has had an unforeseen huge effect on the agricultural output and allowed to produce more while using fewer resources (Figure 5). Mankind had escaped the predicted Malthusian catastrophe thanks to advanced technology and engineering. This was an important lesson learned in the history in which technology made the difference.

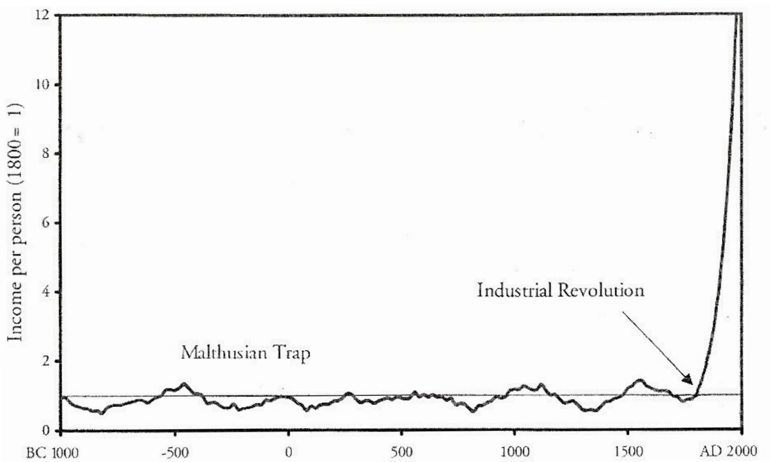

Figure 5. World economic history and the escape from Malthusian Trap because of Industrial Revolutions (adapted from: [7])

From Human Intelligence to Artificial Intelligence

With the progress in engineering and herewith the triggering of the first and second industrial revolutions, mankind has increased its impact on the environment and society which has eventually led to a dominant position on Earth. In fact, this dominant position is not due to the physical skills but can be primarily attributed to the ingenuity of the human brain. Without the intellect of the human being, advanced engineering, the many technological inventions, the scientific theories and herewith the industrial revolutions would not have been possible at all. It is this ingenuity of the brain that also drives modern civilization and provides the human being a special role on Earth.

A closer look at the human brain shows that it contains more than 100 billion neurons and it is relatively very large with respect to the body size. Compared to other mammalian brains, the human brain is indeed by far the most cognitively able and the smartest. What is more, human brain power is not an eternally fixed constant but can be enhanced [9][10]. This can be achieved with education and training, better infant nutrition, adequate sleep, and exercise. In addition, we can also enhance our *collective intelligence*, a shared or group intelligence that emerges from the collaboration and collective efforts of individuals [11]. Collective intelligence strongly contributes to the shift of knowledge and power from the individual to the collective. Collective intelligence can be improved by improving our epistemic institutions, provide better tools for communication and collaboration and involve more people in the problem-solving process. Enhancing individual and collective human intelligence needs to be pursued to support the solution of societal problems.

Another way to support and enhance brain power is through developing machines that can take over the human activities. As we have discussed before, in the Industrial Revolution several mechanical devices were developed to support the automation of tasks that required physical effort. The invention of an intelligent machine that could take over the human mental activities, the computer, paved the way for the **Third Industrial Revolution**, or the Digital Revolution [12]. This phase started in the late 1950s and can be characterized as a transition from mechanical and analogue electronic technology to digital electronics with the adoption and proliferation of digital computers. A computer is simply a machine that can be instructed to automatically execute a software program, that is, an arbitrary set of arithmetic or logical operations. The software program defines the intelligence and can be written to support a very wide variety of tasks. The first programs were expressed in machine code, a set of instructions executed directly by a computer's central processing unit (CPU). Because each computer had its own specific set of machine language operations, the computer was difficult to program and limited in versatility and speed and size. This problem was solved by assembly languages, which replaced the cryptic binary codes for the computer operations with symbolic notations. Although there was a fundamental improvement over the previous situation, programming was still difficult. Hence, high-level programming language with further abstractions from the details of the computer were introduced in the subsequent years. The first FORTRAN (Formula Translation) compiler was released by IBM in 1957 [13]. The ALGOL (Algorithm Language) compiler (1958) provided new concepts that remain today in procedural systems. LISP (LISt Processor), implemented by McCarty at MIT in 1958 was a language designed for symbolic processing and formed the basis for the functional software programming paradigm. Intended for artificial intelligence programming its earliest applications included programs that performed symbolic differentiation, integration, and mathematical theorem verification.

Software became an important asset for an increasing number of organizations. Moreover, the size and the complexity of software systems also increased dramatically. Soon it became clear that software cannot be developed in an ad hoc manner but a systematic engineering approach is needed instead. The terms *software crisis* and *software engineering* were coined at a NATO conference in 1968 [14][15]. At that time, many software projects were delivered late, over budget and with low quality. To cope with this so-called software crisis, it was thus agreed that a software engineering approach was needed to be able to develop software in time, within budget and with the required quality. A novel engineering discipline was born that would have a huge impact on the global world society.

The advances in computing, including both hardware and software, led also to the idea of *Artificial Intelligence (AI)*. In its broadest sense, AI is defined as an area of computer science that deals with giving machines the ability to seem like they have human intelligence. The field of AI research itself was founded at a conference at Dartmouth College in 1956 [16]. AI flourished in the 1960s, and a lot of funding was reserved for AI and many AI laboratories were established around the world. The confidence in AI was at such a high level that AI researchers made overly optimistic statements [17][18], such as:

- 1957, H. A. Simon: Most theories in psychology will take the form of computer programs.
- 1958, H. A. Simon and Allen Newell: “within ten years a digital computer will be the world’s chess champion” and “within ten years a digital computer will discover and prove an important new mathematical theorem.”

- 1965, H. A. Simon: “machines will be capable, within twenty years, of doing any work a man can do.”
- 1967, M. Minsky: “Within a generation ... the problem of creating ‘artificial intelligence’ will substantially be solved.”
- 1970, M. Minsky: “In from three to eight years we will have a machine with the general intelligence of an average human being.”

Most AI researchers shared this prevailing but early optimism. When the expected predictions were not realized, progress in AI research substantially decreased, leading to the so-called *AI winter*, a period of reduced funding and interest in artificial intelligence research [19]. Two explanations can be given for this misjudgment of AI in its early stage. First, at the time when AI was introduced the available memory and computing power were far behind and had limited capacity that is needed for the intelligence that is at least at the level of that of a human being. Second, at the time when the concept of AI was introduced the concepts related to intelligence were not very understood yet. The AI field draws upon different disciplines including computer science, mathematics, psychology, linguistics, philosophy, neuroscience and artificial psychology. To provide intelligence that is close to or surpasses human intelligence requires a thorough understanding of each of these disciplines and their relations.

For many researchers, AI is still a long way from surpassing human intelligence and there are many open questions that need to be answered before we can fully grasp the notion of intelligence. Despite of the slowdown of the progress, research related to different disciplines of AI has continued. The speed, power, and versatility of computers continued to increase exponentially since the introduction of the computer. A measure for evaluating the progress of computing has been the Moore’s Law which states that the number of transistors on integrated circuits and likewise the performance of processors doubles approximately every two years [20]–[22]. Indeed, since the introduction of the law in 1965, the law seems to have quite accurately described and predicted the developments of the processing power of components in the semiconductor industry.

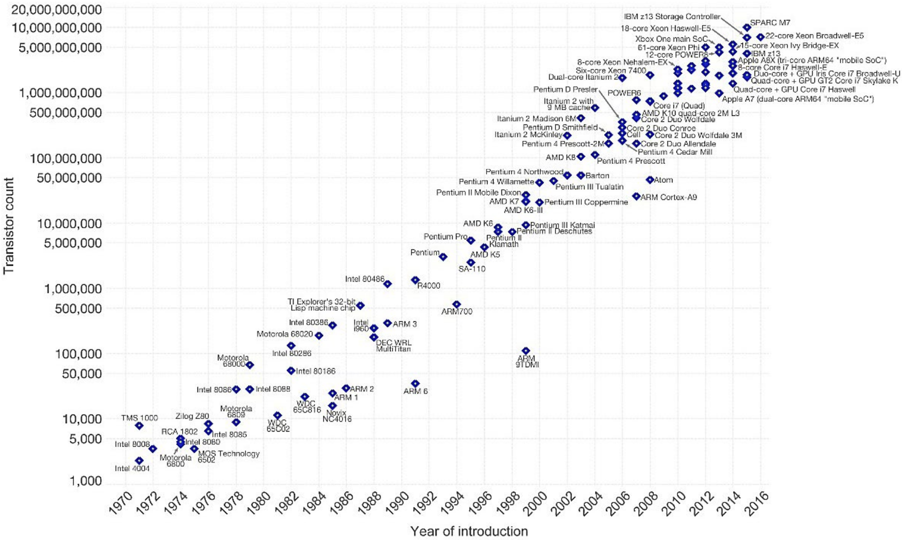

Figure 6. Moore's Law – Number of transistors on integrated circuit chips (adopted from: Wikipedia http://en.wikipedia.org/wiki/Transistor_count)

Currently it is recognized that increasing the processing power of a single processor has reached the physical limitations [20]. Hence, to increase the performance the current trend is towards applying parallel computing on multiple nodes. Here, unlike serial computing in which instructions are executed serially, multiple processing elements are used to execute the program instructions simultaneously. With parallel computing, over the last decade the number of processing nodes has continued to increase exponentially to tens and hundreds of thousands of nodes providing massive processing performance. Parallel computing is applied in supercomputers, a computer with a high-level computational capacity compared to a general-purpose computer [23]. Supercomputers are used for a broad range of computationally intensive tasks in various research and engineering domains, including climate research, quantum mechanics, oil and gas exploration, molecular modeling, physical simulations, testing, life science research, advanced manufacturing, and data analytics. The performance of a supercomputer is measured in floating-point operations per second (FLOPS). Unlike human brain capacity, the capacity of supercomputers has been growing exponentially. The first supercomputer in 1964, the CDC 6600, operated only at a speed of three megaflops (10^6). It took then 21 years to go to gigaflops (10^9) in 1985 with the Cray 2 supercomputer. In 1999, the teraflops-scale computing was reached, or a trillion (10^{12}) FLOPS. In 2008, Los Alamos' Roadrunner supercomputer reached petascale computing, or a quadrillion (10^{15}) FLOPS. As of June 2016, the completely homemade *Sunway TaihuLight*, a Chinese supercomputer is ranked number one in the TOP500 list [24] as the fastest supercomputer in the world, including more than 10 million cores, and a performance of 93 petaflops [25]. Interesting to know, the computer costs $400 million, and burns about 20 megawatts of electricity, enough to power 20,000 households.

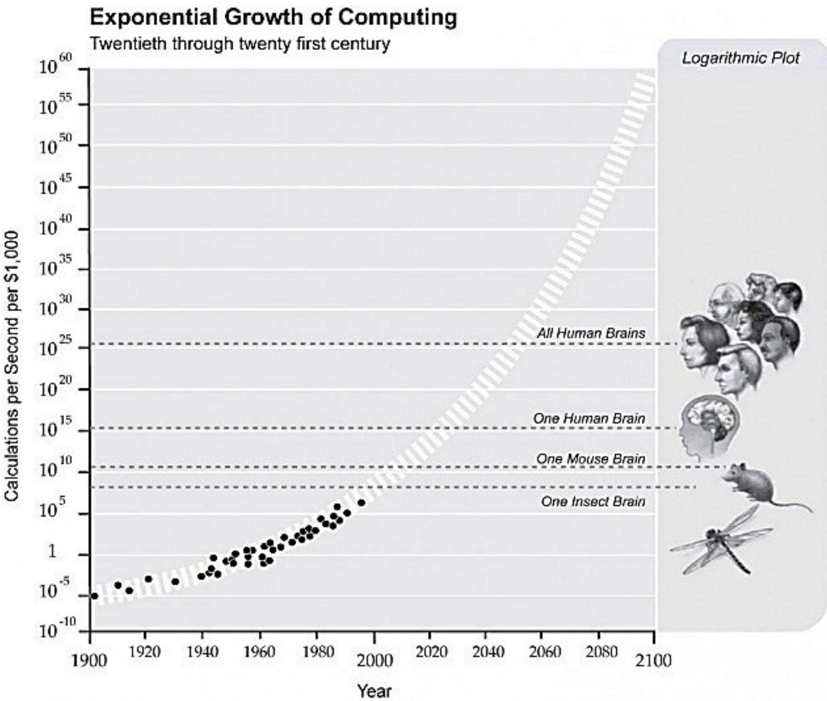

Figure 7. Exponential Growth of Computing and Artificial Intelligence – each dot in the graph is a computer (adopted from: singularity.com)

The exponential growth of computing in recent decades was realized with integrated circuits, first by using sequential computing, later with parallel computing. It is stated that this exponential trend in computing began well before Moore noticed it in integrated circuits. According to the futurist Kurzweil [26], the Moore's Law is the so-called fifth computing paradigm. The first four include computers using electromechanical, relay, vacuum tube, and discrete transistor computing elements. It is expected that the supercomputers of the future will be based on three-dimensional molecular computing or quantum computing [27][28], the next and sixth paradigm. This will further exponentially increase the computing power in an unforeseen way.

Despite of these developments, as of today, the human brain is still far more advanced and efficient than the supercomputers that have ever been built. The human brain reportedly can store about 2.5 petabytes of memories. This can already be achieved by current technology. However, the speed of processing information is, for the time being, a challenge. The human brain contains on average about 100 billion neurons (10^{11}) and about hundred trillion synapses (10^{14}). Each neuron can fire about 100 times a second. Based on these facts, it is postulated that the human brain operates at 1 exaFLOP (10^{18}), which is equivalent to a billion times billion calculations per second. Nevertheless, it is expected that the first supercomputer that can reach exascale computing, that is, 10^{18} FLOPS will appear soon after 2018 (Figure 7). If we consider the computers of only two or three decades ago (Figure 8), we can conclude that computing has progressed exponentially.

Figure 8. Example Advertisements of Computers and Components in the 1980s

Given the high progress in computing the case for AI has now regained interest [29]. Could we indeed realize the initial goals of AI, that is, an intelligence exhibited by machines? To answer this question, we need to take a close look at the scope and the categories of AI. In principle, we can distinguish three different AI categories to achieve machine intelligence:

Artificial Narrow Intelligence (ANI), also referred to as *weak AI* relates to machine intelligence that equals or exceeds human intelligence for a particular domain. Examples include ANI for games such as Chess, Checkers, Scrabble, Backgammon, self-driving cars, smartphone, e-mail spam filter, the Nest thermostat, and Google

Search, just to name a few. Deep Blue was the first computer chess-playing system to beat a reigning world chess champion, Garry Kasparov on 11 May 1997 [30]. In a Jeopardy! quiz show exhibition match, IBM's question answering system, Watson, defeated the two greatest Jeopardy champions by a significant margin [31]. In March 2016, AlphaGo won 4 out of 5 games of Go in a match with Go champion Lee Sedol [32]. ANI has been proven for various domains and it is expected that the number of domains in which ANI is applied will grow further.

Artificial General Intelligence (AGI), or *strong AI*, Human-Level AI refers to a computer that is as smart as a human being and can do intellectual tasks that a human being can. Since AGI includes a wider set of domains it is indeed more difficult to create these than ANIs. The famous Turing test is a test of a machine's ability to exhibit intelligent behavior equivalent to, or indistinguishable from, that of a human [33].

Both ANI and AGI aim to achieve the same level of intelligence of a human being, either in a special domain or in general. Some researchers believe that artificial general intelligence will be shortly followed by artificial superintelligence. **Artificial Superintelligence (ASI)** refers to "an intellect that is much smarter than the best human brains in practically every field, including scientific creativity, general wisdom and social skills."[10]. The degree of intelligence in ASI can range from just a little smarter than a human to one that's billions and trillions of times smarter. In principle, if sufficiently intelligent software is produced, it would be able to reprogram and improve itself, after which it would be even better at improving itself, and this would continue in a rapidly increasing cycle. In the end this would lead to an *intelligence explosion*, and we would have a so-called *superintelligence*. The reason behind this is the so-called *Law of Accelerating Returns* as proposed by the futurist Ray Kurzweil [26]. Hereby, it is assumed that more advanced entities can progress at a faster rate than less advanced entities. Because of this law, Kurzweil believes that the 21st century will achieve 1000 times the progress of the 20th century. As a result of this, superintelligence will far exceed human intelligence, and the intelligence difference between the normal and smartest human beings would be just negligible (figure 9).

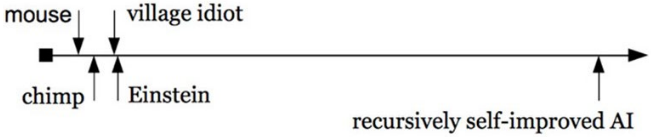

Figure 9. Scale of intelligence (adapted from: intelligenceexplosion.com)

Smart Connected Products

Whether we will indeed have an intelligence explosion and reach superintelligence is one of the interesting philosophical debates in artificial intelligence [34]. Nevertheless, what is clear is that computer performance has substantially increased the last decades, and we are now able to develop smart products that can take over or support the human beings in performing intelligent tasks. We can observe a massive increase of digitalization, that is, the application of digital technology in all aspects of human society. Computing is now not only in computers but can be anytime and everywhere. Based on the emphasis, this paradigm of computing is often called *ubiquitous computing* [35], *pervasive computing*, or *ambient intelligence* [36]. In this paradigm, the ordinary things become "thinking things" or smart products. Simply speaking smart systems are products or things in general that include smart hardware components, software components and connectivity to provide smart connected behavior.

In essence, the capabilities of smart products can be grouped into four areas: *monitoring*, *control*, *optimization*, and *autonomy* [37][38]. Each capability builds on the preceding one, control requires monitoring, optimization requires control and monitoring, and autonomy requires all the three. Monitoring implies the observation of a system's condition, operation, and external environment through sensors and external data sources. Control implies the regulation of systems through remote commands or algorithms that are built into the device or exist in the cloud. The data collected from the monitoring activity together with the control capability allows the optimization of the system. Optimization can be implemented using dedicated algorithms and analytics that can adopt the monitoring data optionally with the historical data. The goal of optimization is typically to improve the quality of the system including its effectiveness and efficiency. The highest level of smart behavior is *autonomy* that combines monitoring, control, and optimization to learn about the environment, self-diagnose the own goals and needs, and adapt to the changing preferences (Figure 10). An autonomous system may be interfaced with other systems but it controls its own actions.

Smart systems can also be connected to and act in coordination with other smart systems to create a connected intelligence. This connectivity of smart products has two key purposes. On the one hand, it enables the communication with other smart products and systems. On the other hand, it allows one or more of the required capabilities to exist outside of the physical device, that is, the product cloud. The product cloud can include additional services for data storage, data analytics and smart software applications for supporting the monitoring, control, optimization and autonomous operations of the product functions.

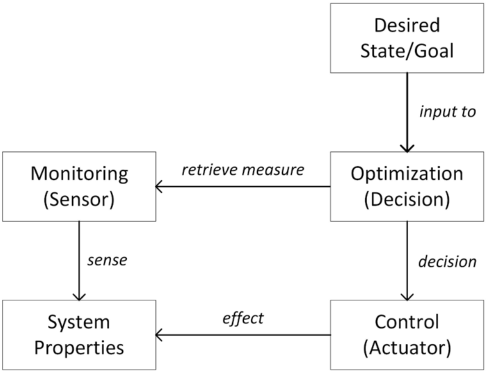

Figure 10. Model for a Smart System

The interconnection of smart products over the internet is defined as the *Internet of Things (IoT)* [39][40][41]. In fact, the introduction of the internet starting in the 1980s was an important milestone in the third industrial revolution allowing the communication between people at any time and any place. With the internet, the different business activities could now be better coordinated and integrated across different geographical locations. For example, organizations in globally distributed supply chains could now be closely integrated. Different from the traditional internet which enabled the connectivity of human beings, the IoT enables anytime, anyplace connectivity not only for human beings but also for anything.

In the IoT world, physical things and virtual things, all interact with each other in the same space and time. The things in IoT are not passive devices anymore, but smart devices with capabilities of communication, sensing, actuation, data capture, data storage and data processing. Typically, the devices will collect different kinds of information and provide it to the information and communication networks for further processing. With the IoT the world has already deployed about 5 billion "smart" connected things. Predictions say there will be 50 billion connected devices by 2020 and in our lifetime, we will experience life with a trillion-node network.

The IoT is one of the main triggers for the *big data* paradigm that provides innovative opportunities and value for businesses which was not foreseen or possible before, due to the limited memory and computing power [42]. Big data is often characterized by four Vs': volume (amount of data), variety (range of data types and sources), velocity (speed of data in and out) and veracity (quality of captured data).

The ability to connect smart products that can communicate with each other is leading to a proliferation of *smart systems*. An example of a smart system is home automation or smart home which involves the control and automation of lighting, heating, ventilation, air conditioning, security and home appliances such as washers, ovens or refrigerators, and coffee machines. Other examples of smart systems include smart cars, smart office, smart hospital, and smart farming. The digitalization is massive and invasive.

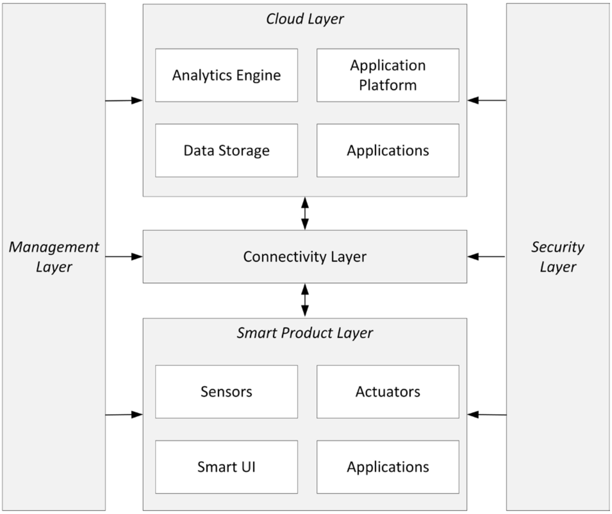

Figure 11. Architecture for Connected Smart Systems

Various reference architectures have been provided for the IoT. In general, IoT architecture is represented as a layered architecture with various set of layers (Figure 11). Hereby, a layer simply represents a grouping of modules that offers a cohesive set of services. The *Smart Product Layer* includes the capabilities for the smart products in the network. The *Connectivity Layer* provides functionality for networking connectivity and transport capabilities. The *Cloud Layer* consists of functionality for generic support capabilities (such as data processing or data storage), and specific support capabilities for applications. The *Security Layer* is a side-car layer relating to the other four layers, and provides the security functionality. Finally, the *Management Layer* supports capabilities such as device management, local network topology management, and traffic and congestion management.

Industry 4.0

Obviously, the world as we know and experience it today has been shaped by the earlier described three major industrial revolutions. With the first industrial revolution, the mechanization of production was mobilized using water and steam power. The second industrial revolution resulted with the help of electric power in mass production. Finally, the third revolution or digital revolution resulted in further automation with the use of electronics and IT. Similar to the first two industrial revolutions the third industrial revolution had a disruptive impact on the world in which we live and work.

The rapid progress and evolution of computing with smart devices that can be connected over an advanced IT infrastructure with internet services has triggered the transition to the **Fourth Industrial Revolution**, or **Industry 4.0** [43][44]. Interestingly, although the first three industrial revolutions were observed afterwards, the fourth industrial revolution which is starting to happen is being predicted a-priori. Further, the period between the different industrial revolutions have become shorter, indicating the rapid technological developments (Table 1).

The term Industry 4.0 was first coined by a German initiative in 2011 that aimed to promote the idea as an approach to strengthening the competitiveness of the German manufacturing industry [45][47]. Related overlapping terms are Smart Factory, Smart Industry, the Industrial Internet of Things (IIoT), or Advanced manufacturing.

Table 1. Industrial Revolutions

Industrial Revolution	Time Period	Adopted Technology and Capabilities
First Industrial Revolution	1784-mid 19th century	Mechanical production with the help of steam and water power
Second Industrial Revolution (Technological Revolution)	Late 19th century to 1970s	Mass production with the help of division of labour and electrical energy
Third Industrial Revolution (Digital Revolution)	1970s – Today	Automated production with the use of electronics and information technology
Fourth Industrial Revolution (Industry 4.0)	Today-	Mass customization with the use of cyber-physical systems
Fifth Industrial Revolution	around 2050?	Superintelligence?

In fact, the term *Industry 4.0* refers to the combination of several major innovations in digital technology including artificial intelligence, advanced robotics, cloud computing, the Internet of Things, big data capture and analytics, digital fabrication (including 3D printing), mobile devices, and the embedding of all these elements in an interoperable global value chain. These technologies are coming all to maturity right now but are often thought of separately. But when they are applied together, they integrate the physical and virtual worlds and as such the new world with industry 4.0.

Industry 4.0 relates to the context of production and implies that manufacturing environments will comprise smart machines, storage systems and production facilities capable of autonomously exchanging information, triggering actions and controlling each other independently. As such it elaborates on the developments as we have seen in the first three industrial revolutions. Industry 4.0 creates a cyber-physical production system (CPPS) or a "smart factory" [48]. Hereby, cyber-physical systems monitor physical processes, communicate and cooperate with each other and with humans in real time, and make decentralized smart decisions. Smart factories also use smart products, that are uniquely identifiable, and know their own history and current status, and can even support the production process.

Within Industry 4.0, the boundaries of individual factories and systems in general will largely cease to exist and will be lifted to interconnect multiple systems or even geographical regions. With the increased interconnection of the production systems the complexity of production and supplier networks will grow enormously. By connecting smart machines and smart factories, businesses can create intelligent networks along the entire value chain. Smart factories will heavily use information and communication technology and trigger the further evolution in the supply chain and production line with an advanced level of both automation and digitization. Within smart factories, smart machines will interact with each other using smart machine-to-machine (M2M) communication. Advanced AI reasoning, self-configuration, self-diagnosis, and self-optimization will be applied to complete complex tasks to deliver vastly superior cost efficiencies and better quality goods or services.

Industry 4.0 focuses on end-to-end digitization of all physical assets and the integration into digital ecosystems with value chain partners. Companies that adopt Industry 4.0 will also benefit from mass customization, that is, the inclusion of customer-specific criteria in the design, configuration, ordering, planning, manufacturing and operation of the products.

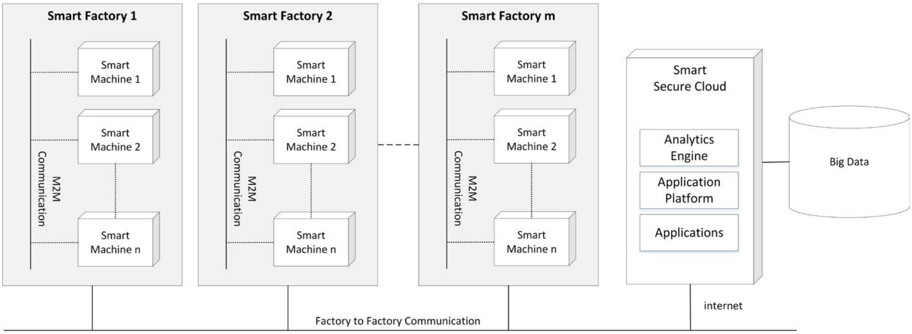

Figure 12. Integrated Smart Factory System

System-of-Systems Engineering

With the current development of smart connected products, the scope of systems has broadened as well. Whereas traditionally systems were addressing a single product or domain, we can now observe the composition of multiple smart systems that are integrated in a coherent way, leading to a *system-of-systems (SoSs)* [49]. A *system-of-systems* is an arrangement of systems that results when independent and useful systems are integrated into a larger system that delivers unique capabilities [23][24][25]. In an SoS, the individual system components have *managerial* and *operational independence*, whereas the overall purpose of the system is to provide a function or service that cannot be provided by the individual systems independently [52]. The concept of system and system-of-systems is generally applicable to different categories of systems including [53]:

- *Technological Systems*: include man-made engineered artifacts or constructs; including physical hardware, software and information.
- *Social Systems*: include elements, either abstract human types or social constructs, or concrete individuals or social groups.
- *Natural Systems*: include elements, objects or concepts which exist outside of any practical human control.

Besides of the above classification, in practice, an SoS can also consist of multiple different types of elements and form a *hybrid SoS*. An important category hereby are systems which contain both technical and social elements, that is, human elements. The behavior of such *socio-technical systems* is determined both by the nature of the technical elements and by their ability to integrate with or deal with the variability of the natural and social systems around them.

Table 2. Example System-of-Systems

Type	System	System-of-Systems
Technological	Airplane	Airport, Air Traffic Control System
	Car, Road	Integrated Traffic System
	Train	Train Station, Rail Network
	Smart Metering, Wind Turbine	Smart Grid
	Computer	Distributed System, Software Ecosystem
	Farm	Integrated Precision Farming System
	Building	Town, Shopping Mall
Social	Town Council	Government, United Nations, European Union
	Family, Social Group	Town, Nation
	Student, Teacher, School	Education System
	Company	Enterprise, Stock Market
Natural	Animal	Herd
	Plant	Forest
	Weather, RI	Eco-system
	Star	Solar System

From the historical perspective, the focus on SoSs is an important milestone for engineering which has primarily based on single disciplines such as mechanical engineering, electrical engineering, and software engineering. Within the context of technical systems, *Systems of Systems Engineering (SoSE)* focuses on designing, analyzing, implementing and maintaining such large so-called systems of systems (SoS). Traditionally, SoSE methodology is heavily used in the defense domain but is now also increasingly being applied to non-defense related problems such as architectural design of problems in air and auto transportation, healthcare, global communication networks, space exploration and many other SoS application domains including the life sciences domains (Table 2).

In general, systems in SoSs are being employed in various combinations to provide different capabilities. Different types of SoS can be distinguished based on different properties including central management of the systems, a common agreed purpose and the independent ownership of the systems (Figure 13). *Directed SoS* are built and managed to fulfill specific purposes. The SoS is centrally managed to fulfill the agreed purpose. The component systems can operate independently, but their normal operational mode is subordinated to the central managed purpose. *Acknowledged SoS* have also a recognized objective and central management. In contrast to Directed SoS the constituent systems retain their independent ownership.

Changes in the systems are based on collaboration between the SoS and the system. *Collaborative SoSs* have a commonly agreed purpose but do not have a central management. The component systems interact voluntarily to fulfill agreed-upon central purposes. *Virtual SoSs* lack a central management authority and a centrally agreed-upon purpose. Further the component systems have their independent ownership.

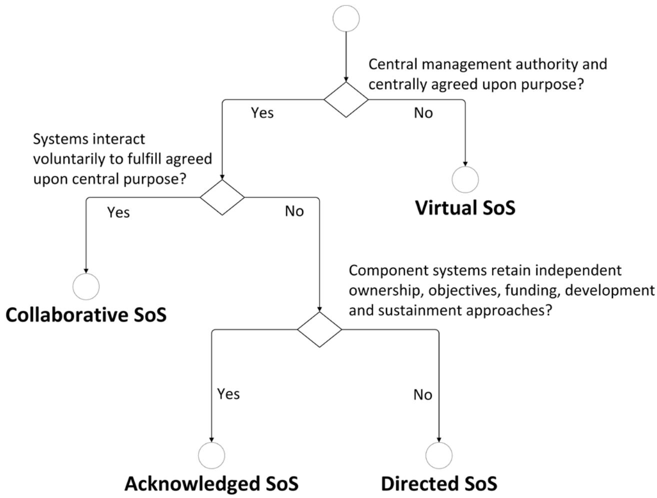

Figure 13. Different types of SoSs

In practice, SoSs can be defined in different ways and include many different features [54][55]. Some features such as operational and managerial independence seem to be common for all SoSs. Other features are variable such as the number of owners or the design of the substituent systems (Figure 14).

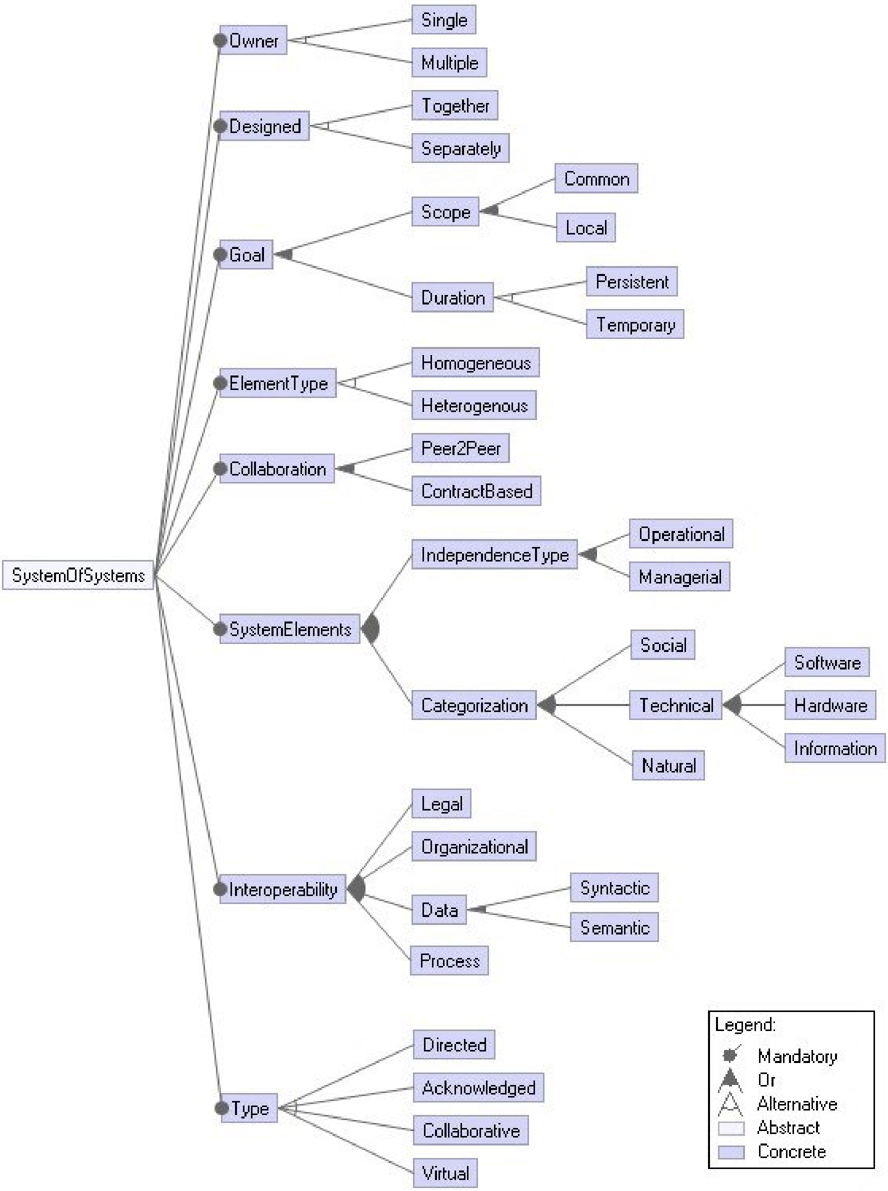

Figure 14. Feature-Diagram of System-of-Systems

A typical SoS is a supply chain system [56]. A supply chain is defined as a system consisting of organizations, people, activities, information, and resources involved in moving a product or service from supplier to customer. Supply chain activities transform natural resources, raw materials, and components into a finished product that is delivered to the end customer [57]. Due to the increased global competition, many companies are forced to improve their efficiency of the supply chain using systematic supply chain management (SCM) approaches. The underlying idea for SMC is based on the observation that practically every product that reaches an end user represents the cumulative effort of multiple organizations defining the supply chain. Supply chain management, as such, is the active management of supply chain activities to maximize customer value and achieve a sustainable competitive advantage. SCM activities typically include the management of the flow of materials, information, and finances in a process from supplier to manufacturer to wholesaler to retailer to consumer. Further, SCM involves coordinating and integrating these flows both within and among companies. To manage and optimize the material and information flow that propagates up the supply chain from the source of demand to the suppliers it is important to consider the SCM as an SoS. In general, the separate systems in the chain are not designed and developed together. Supply chains that do not have a central management can, as such, be characterized as *collaborative SoSs*. However, currently many supply chains need to conform to common standard regulations and guidelines to meet the global requirements such as for example transparency. In this case the SCM could be characterized as an *acknowledged SoS*. A further observation is that an SCM is heavily based on IT and we can state that the goals of a current demand-driven SCM cannot be achieved without the proper integration with IT systems. We can already identify the following important components in a SCM system:

- *Enterprise resource planning systems (ERP)* – providing services for purchase management, production management and sales management, in particular for manufacturers and trading companies;
- *Warehouse Management Systems (WMS)* – providing services for logistics, in particular for wholesalers;
- *Transport Management Systems (TMS)* – providing services for transport booking, planning and monitoring.

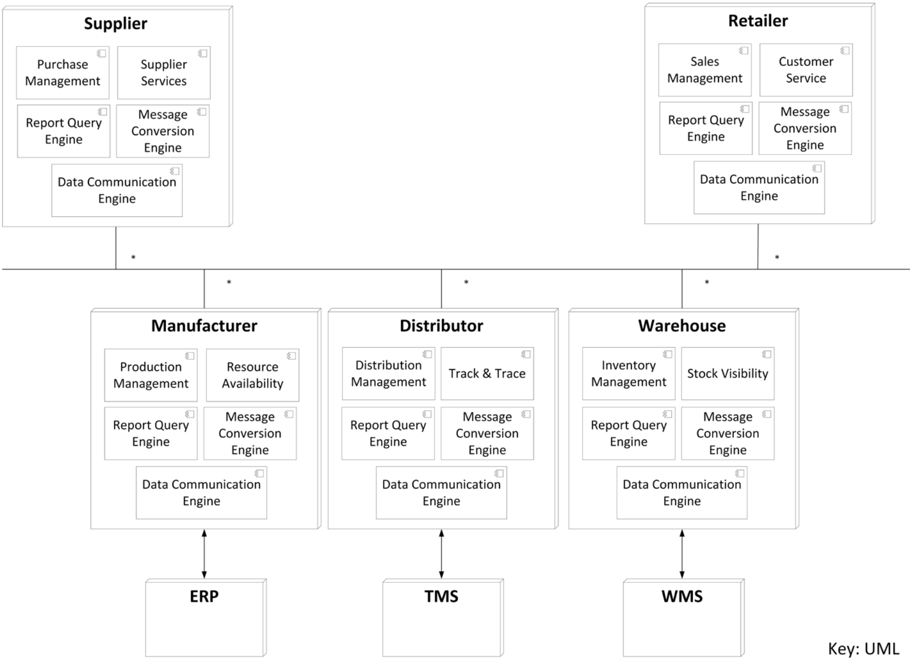

Figure 15. Supply Chain Management System Design – An example of a System-of-Systems

Figure 15 shows the overall architecture of the SCM with the current IT components [58]. The architecture consists of five different node types: Supplier, Manufacturer, Distributor, Warehouse and Retailer. Note that each node has three similar components including Report Engine, Message Conversion engine and Data Communication Engine. Further each node has also its' specific type of components. Finally, a manufacturer node is connected to an ERP system, a distributor to a TMS and a warehouse to a WMS. Besides of these traditional information systems we can also observe the integration of recent developments such as cloud computing, IoT, big data analytics, smart systems, and Industry 4.0. The data for the supply chain is often retrieved from smart things. This data is stored and processed on the cloud, and decision making is supported through advanced big data analytics. Every system in the supply chain can then be considered as a smart system which is connected with other smart systems in the chain.

Challenges for System-of-Systems Engineering

Clearly, engineering connected smart system-of-systems implies a holistic multidisciplinary approach that usually aims to integrate data, software, hardware, people and processes within the organizational, institutional and societal context. Even though single system engineering techniques seem to be on first sight applicable for SoS, they do not appear to scale reliably to SoS. We can identify the following important concerns and challenges for engineering SoSs [59]:

Decentralized Engineering

In traditional single system engineering the focus is on adopting a top-down engineering approach that goes through a requirements elicitation to the design, and implementation of the system. An SoS is usually at a larger scale and has a broader scope than traditional systems. As such, the traditional central engineering perspective is no longer adequate. Because of the operational and managerial independence of the substituting systems, an SoS necessarily requires a decentralization of the engineering process including decentralized data, decentralized development, decentralized evolution and maintenance, and decentralized operation control. In this context, while developing systems can be compared to developing buildings, developing SoSs can be more compared to developing cities. Cities are system-of-systems in which the system components are in fact not centrally engineered but gradually developed and regulated. To cope with this issue the traditional engineering approaches which largely focus on central development need to be reconsidered to align with the scale and concerns of SoSs.

Continuous Evolution and Deployment

SoSs do not only constitute multiple system components but will also be in service for a longer time. Unlike single systems it will be impractical to replace or retire SoSs altogether. Instead SoSs will be characterized by an increasing need for evolution whereby new capabilities are deployed, and unnecessary capabilities will be given up. The analogy can here be again given of the evolution of cities. Cities must continuously function despite ongoing maintenance and improvement activities. The evolution of SoSs is also continuous, and adaptations will be made continuously to meet the changing requirements. Further, for the adaptation process it will be necessary which changes have only a local impact and as such can be carried out concurrently, and which changes have systemic impact over the whole SoS and as such must be coordinated. To embrace change and allow corrections of a system, different development processes have been proposed including iterative, incremental and agile approaches [60][61]. Further DevOps (a clipped compound of development and operations) has been defined as a practice that emphasizes the collaboration and communication of both software developers and other information-technology (IT) professionals while automating the process of software delivery and infrastructure changes [62]. Despite of these developments current approaches do not scale yet to cope with the challenges of continuous evolution and deployment that we can observe in SoSs.

Socio-Technical Concerns

In SoS, people are not just users of the system but will be an active part who design, develop, use, test and maintain the system. Hence, it will be hard to understand the aspects of an SoS without full consideration of the human behavior in the system. When designing current systems, the focus tends to be on technology, and the functional and quality concerns of the technical system. A socio-technical perspective that takes into account both the technical and the human aspects is largely missing. Hereby, not only the behavior of individuals but also the collective behavior of groups of users and developers will need to be analyzed to get an insight in how the SoS is used, viewed, accepted, and maintained. A socio-technical system typically can be considered as consisting of a social subsystem and technical subsystem (Figure 16) [63]. The social subsystem consists of people and people in relation to each other (i.e. structure). The technical system does not include human elements and consists of technology and process. The process component defines the business process, that is, the series of steps to complete a business activity. The technology component of the technical system consists of software, hardware and the networking or telecommunications. A problem in socio-technical systems is the alignment of these components. The so-called business-IT alignment problem has been addressed largely for single systems but not yet explicitly considered at the scale of system-of-systems [56].

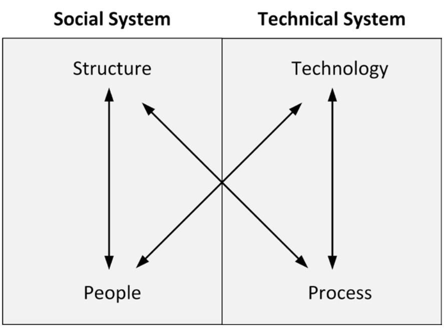

Figure 16. Model for Socio-Technical System

System-of-Systems Ecosystem

An SoS is constituted of dynamic environment of a set of interdependent sub-systems that comprise computing devices, people and organizations. In this context, we can consider many SoSs as *socio-technical ecosystems*. The concept of ecosystem is inspired from natural ecosystems in which organisms are characterized by symbiotic relationships and their survival relies heavily on the survival of the ecosystem. In the domain of software systems, a software ecosystem (SECO) is a collection of systems, which are developed and co-evolve in the same environment [64]. Typically, a SECO consists of a common software platform and a community of internal and external actors that compose software systems to satisfy their needs. SECO based development is different from the traditional software development in which a software product was the result of effort of an independent software vendor typically developing a monolithic product. In software ecosystems, the development is not intra-organizational but inter-organizational and as such spread outside the traditional borders of software companies to a group of companies, private persons, or legal entities. Software ecosystems is also different from traditional outsourcing techniques since the initiating actor or platform owner does not necessarily own the software produced by contributing actors. To develop socio-technical SoSs it is important to integrate the concepts of SECO. Here it should be noted that current SECO architectures are typically focused on a single platform system with multiple developers and consumers. This does not align with the larger size and scope as it is required from the SoS.

Multi-Paradigm Modeling and Engineering of SoS

SoSs are complex and dynamic systems that integrate physical, software, and network aspects. To date, no unifying theory nor systematic design methods, techniques and tools exist for such systems. Individual (mechanical, electrical, network or software) engineering disciplines offer dedicated solutions for systems in their disciplines but are limited when considering SoS. To support the engineering and integration of SoSs, a multi-paradigm modelling (MPM) approach is needed to model the SoS at the appropriate level(s) of abstraction. MPM is a research field that aims to combine different levels of abstraction and views, using modeling formalisms and semantic domains, with the goal of simulating or realizing systems. The key challenges in MPM are finding adequate modeling abstractions, multi-formalism models, and model transformations. For developing SoS multi-paradigm modeling will be necessary.

Design Optimization

SoS is a collection of task-oriented or dedicated systems that pool their resources and capabilities together to create a new, more complex system which offers more functionality and performance than simply the sum of the constituent systems. To provide the global level optimization while considering the local optimizations of the constituting systems in SoS, novel design optimization approaches are required. The optimization of the configuration will need to be dynamic due to the adaptable and open-ended behavior of SoS. Different quality factors need to be identified and a trade-off analysis performed to constantly tune the SoS to a feasible configuration [65][66]. In particular, for the case of directed and acknowledged SoS in which tasks are allocated over different systems it is important to search for the proper configuration to achieve the SoS level quality concerns.

Parallel Computing of SoS

SoS consists of various systems that do not execute serially but inherently run in parallel to each other because of the operational independence. This situation does not only help the natural modelling of the real world but also offers opportunities for distributing and allocating tasks to multiple nodes in different systems to increase speed-up and efficiency. Parallelism of algorithms and tasks for well-defined domains has been broadly addressed in the literature. However, parallelism in the context of SoS that consist of multiple various types of socio-technical elements has not been explicitly addressed. This situation provides a challenge for existing parallel computing models and methods, design abstractions, and the interconnection network for parallel computing in SoSs.

Emergent Properties

The behavior of an SoS is not localized to any component system. An SoS has often to deal with emergent behaviors that can appear when several system components interact in complex ways. Emergent behavior is only exhibited at the global SoS level and cannot be achieved by any of the constituent systems. The SoS will need to have capabilities to respond to emergent behavior and, as such, must be able to observe their own operations, recognize acceptable and unacceptable behaviors, and take corrective action with little or no operator intervention [55]. On its turn this will require the fundamental models of SoS that are linked to the conceptual notion of emergence. In addition, corresponding tools will be needed through which operators can gain early warning of potential emergent behavior and devise strategies to deal with it.

Governance of SoS

Different forms of control can be applied in the acquisition and operation of the constituent systems of a SoS. The earlier defined four types of SoS imply also different kind of governance structures. To achieve effective performance of an SoS it is important that the proper organizational structure is selected and applied. Lack of performance of an SoS can be often related to non-technical governance such as mismatches between the organization structures and the global business requirements of the SoS.

Heterogeneity and Interoperability

An SoS is typically a system consisting of multiple independent and heterogeneous systems. In SoS, the elements will be heterogeneous in part because they will come from a variety of sources. The independent system components will run on different hardware/software platforms, developed using different languages and designed according to different methodologies. Hence, an important concern of SoS system design, construction, and evolution is interoperability which will entail integrating heterogeneous elements and engineering perspectives. In SoS interoperability must be addressed and managed at several levels. At the technical level interoperability, will need to address the conventional syntactic interoperability and semantic interoperability. SoS requires however a broader consideration of interoperability beyond the technical level and also consider the social, organizational, and legal level to support the integration of data and processes. For this, an infrastructure will be needed that can combine development, deployment, and operational support for interoperability between organizational teams.

Evaluation of SoS

Evaluation of an SoS will be different from the evaluation of a single system. Existing verification and validation approaches will not be useful due to the scale, heterogeneity and the continuous evolution of SoSs. Moreover, socio-technical SoSs that have human beings as participants require the evaluation not only of technical aspects but also the aspects from the social, organizational and business perspectives. Approaches like agent-based modeling [67] can provide a suitable computational model for simulating the actions and interactions of the different entities in SoSs and for assessing the emergent behavior that is usually a characteristic of SoSs.

Ultra-Large Scale Software Engineering

Even though systems comprise far more than software, it is software that fundamentally makes possible the required intelligence in smart systems. It is mainly for this reason that an increasing number of major businesses and industries are now dependent on software. Most of the products today either consist of software or have been developed using software. Many innovative transformations in society have also been triggered by advances in software technology. Hence, with the increasing need for smart systems, software engineering will be a crucial core competence for most organizations. It will not be possible to develop smart systems without knowledge on software engineering. Broadening the scale to the level of system-of-systems will impose further challenges on the engineering of software. Each of the characteristics of SoS that I have described before implies novel challenges in the fundamental assumptions that underlie the conventional software engineering approaches. Due to the changing scope and scale, current software engineering approaches for developing, deploying and operating software-intensive SoS will not suffice [59].

Escaping the Neo-Malthusian Crisis

So far, I have discussed how mankind has advanced the problem-solving approach over the centuries, which has led to an advanced mature engineering approach. The advanced engineering practices were the primary triggers for new artefacts that resulted in the four Industrial Revolutions. The first three industrial revolutions were primarily the result of mechanization, electricity and IT. As we have discussed before the First Industrial Revolution and as a follow up the Second Industrial Revolution helped to avoid the predicted Malthusian Catastrophe [8]. Since then, however, the world population has grown dramatically, and we are facing a similar situation of about two centuries ago. This "Neo-Malthusianism" view contends that overpopulation may again increase resource depletion or environmental degradation to a degree that does not seem sustainable [32][33][70].

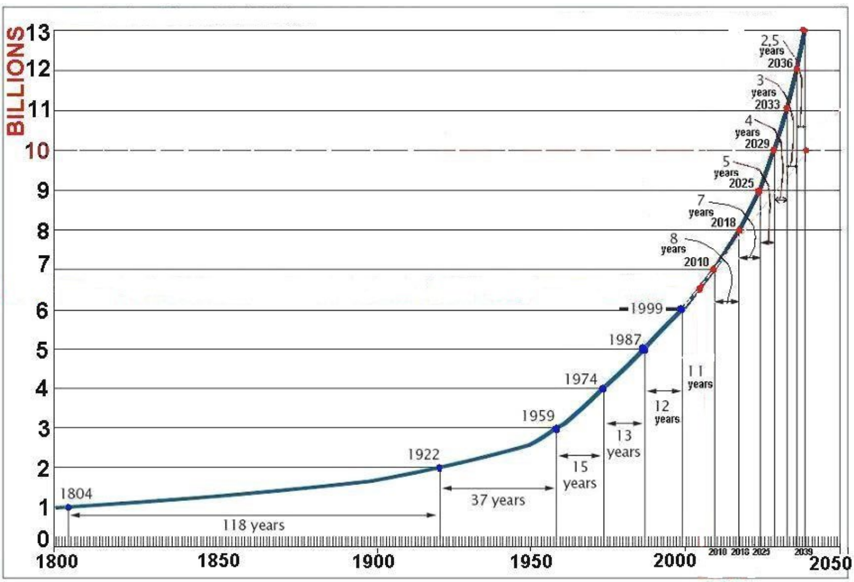

Figure 17. Total World Population. Source: United Nations, Department of Economic and Social Affairs, Population Division (2015). World Population Prospects: The 2015 Revision. http://esa.un.org/unpd/wpp/

The reasons for this worrying situation seems to be justified if we look at the current numbers and estimations of the world population. It is estimated that the world population reached one billion for the first time in 1804 [71]. It was another 123 years

before it reached two billion in 1927, but it took only 33 years to reach three billion in 1960. Thereafter, the global population reached four billion in 1974, five billion in 1987, six billion in 1999 and, seven billion in early 2012. Interestingly, from 1850 till now, the world population increased by about 6 times. According to current projections of the United Nations [71], the global population will reach eight billion by 2024. By 2050 the world will be inhabited by 9 billion people (Figure 17) from which 70 percent will be urban (compared to 49 percent today). To feed this larger, more urban population, it is required that the food production must increase 70 percent. And all this must be achieved despite the limited availability of arable lands, the increasing need for fresh water, and the impact of climate change.
The world is in a precarious position and there are some undisputable reasons to be worried. However, as we have seen from the earlier Malthusian predictions we need to take into account the technological developments that can have a diminishing or reversing impact on the overall pessimistic scenario. Like the disruptive role of engineering in the First Industrial Revolution which helped to break out of the Malthusian Trap, I believe that Industry 4.0, or the Fourth Industrial Revolution, together with the broader socio-technical SoSs, has the potential to help to avoid the Neo-Malthusian catastrophe. If we manage to overcome the challenges for developing smart connected systems engineering, we can optimize food production, manufacturing, energy and resource usage, and provide sustaining solutions for the several grand challenges. Hence, the practice of smart system engineering and herewith the technical developments is not an issue that can be waved aside. On the contrary, before it is too late, more research and investments are needed to solve the challenges for smart system of systems engineering.

Word of Gratitude

As a Turkish migrant child, I was raised in the East of The Netherlands, and have studied and worked for a long time in Twente. I have left now more than 40 years behind me. It was a long road indeed, with all the joys, the challenges and the ongoing life lessons that helped me to advance my thoughts and my viewpoints, both in my professional and my personal life. I am one of the six sons of a farmer who later became a factory worker in a country, far away from home. Twenty-five years ago, my father passed away after a long period of illness. Hence, for me this day is also a day of remembrance to my father, and like my father to all the other many migrant fathers and mothers who worked so hard to contribute to their new homeland. Be kind and thankful, work diligent and never give up, be sincere and down to earth, be polite, don't hurt people love people, be confident and proud on what has shaped all of your identity, build don't destruct, follow the golden rule, don't calculate much in your actions; do the right things for the right reasons. These were the advices of my own parents, which I must say were the primary driving forces for wisdom and success, and which made us what we are now. I am so grateful for this.

I am now at Wageningen University for more than two years now. Since my first day I have been given the feeling that I am *affiliated with*, and not only just work for this university. Many persons have contributed to that embracing feeling and helped me in the whole process in the best possible way.

First, I would like to thank the former rector Martin Kropff and our current rector Arthur Mol for their confidence in me. Next I want to thank our previous director of the department of Social Sciences, Laan van Staalduijnen, and Inge Grimm for the useful talks and advices to help me manage and further build the Information Technology group that I am chairing. Special thanks go to Jack van de Vorst, our current director and the former interim manager of the Information Technology group, for his professional and sincere coaching both during my starting period and the period after.

Thanks to Jacqueline Bloemhof, chairholder of the ORL group, who is always a very collaborative and collegial person. Thanks also to Alfons Oude Lansink, chairholder of the BEC group who is always available for the right advice. Also, I thank the former chairholder of the INF group, Adrie Beulens, who was one of the first to inform me about the new chairholder position. Adrie was available and helpful to provide me the necessary advice for the smooth transition.

I thank my colleagues of the INF group: Ayalew, Cor, Gerard, Gert Jan, Huub, Ioannis, Maarten, Mark, Sjaak, and Sjoukje. We have come to know each other much better now, and I think we are facing a bright future. We know, and the rest who doesn't know yet will know soon, Information Technology and the massive digitalization is not a temporary issue, it will persist and get even stronger. We are all aware of this and we worked so hard together the last two years to realize our goals. In this journey, Anne, Gisella, Ilona, Jeannette, Leonie, and Natasja have always provided the cheerful support. Thank you.

We had several collaborations with different groups and persons at Wageningen University, in particular on the topic of Big Data and smart system engineering. I thank Henk Hogeveen, Dick de Ridder, Arnold Bregt, Jan Top, and Sander Jansen, for the nice and inspiring discussions and collaborations. And of course, many thanks to the group members of ORL with whom we have always a nice collaboration.

Recently we formed the Business Science Cluster with the aim to further strengthen the collaborations. Alfons Oude Lansink, Hans Trijp, Jacqueline Bloemhof, and Onno Omta. I am confident that we will have a nice collaboration the coming years.

I would also like to thank Mehmet Aksit who has always been part of my academic journey, first during my MSc and PhD studies as my supervisor and later on in several national and international software engineering projects. His principle-centered viewpoint, idealism, persisting energy and insight has always been an inspiration for me.

In my whole career, I have had the luck to work with many brilliant and dedicated students. So far, I graduated around 50 MSc students, and supervised more than 20 PhD students on different topics in software engineering, system engineering, and information technology. I will not mention them all. But each student is really so unique and so valuable. I would like to thank them all.

There are so many other people whom I could thank and who somehow had a direct or indirect impact on my journey in life. Thank you all for those who made an impact on my professional and personal life.

In the whole chain of gratitude my family of course deserves a special role. I am so grateful to my dear mother, for her continuous and unconditional love. My five brothers, thank you all for supporting me, giving the right advice, and sharing our thoughts and feelings over the years. Finally, my dear life partner, Leyla, my son Irfan and my daughter Asude, you are really so important to me. Thank you so much for your patience and your love.

We are guests in this world. This lecture was about intelligence, and particularly engineering connected intelligence. Like our intelligence, every-thing is relative, every-thing will cease to exist, no-thing will persist, and one day we will leave too. What remains however will be what we leave behind, our thoughts and our acts, our footprints, our designs of our life. Hereby, we should not forget that, in the end, we are somehow all connected, connected spirit, and connected intelligence.

Thank you for your attention. Ik heb gezegd.

References

[1] N. Bostrom, “How Long Before Superintelligence?,” *Int. J. Futur. Stud.*, vol. 5, no. 1997, pp. 11–30, 2006.

[2] C. Alexander, *Notes of the Synthesis of Form*. Cambridge, Massachusetts: Harvard University Press, 1964.

[3] B. Tekinerdogan and M. Aksit, “Integrating the Concept of Synthesis in the Software Architecture Design Process,” *J. Integr. Des. Process Sci.*, vol. 10, no. 1, pp. 45–56, 2006.

[4] OMG (Object Management Group), “BPMN Specification - Business Process Model and Notation,” *Bpmn.Org Website*. 2015.

[5] E. Evans, *Domain Driven Design*. Addison Wesley, 2006.

[6] G. Clark, “The Great Escape: The Industrial Revolution in Theory and in History,” *Africa (Lond).*, pp. 1–77, 2003.

[7] G. Clark, “The Industrial Revolution,” *Handb. Econ. Growth*, vol. 2, pp. 217–262, 2014.

[8] D. V. Glass and P. Appleman, “Thomas Robert Malthus: An Essay on the Principle of Population (1798).,” *Popul. Stud. (NY).*, vol. 30, no. 2, p. 369, 1976.

[9] N. Bostrom, “Existential Risks,” *J. Evol. Technol.*, vol. 9, pp. 1–28, 2002.

[10] N. Bostrom, “Superintelligence: Paths, dangers, strategies.,” *Superintelligence: Paths, dangers, strategies.* 2014.

[11] A. W. Woolley, I. Aggarwal, and T. W. Malone, “Collective Intelligence and Group Performance,” *Curr. Dir. Psychol. Sci.*, vol. 24, no. 6, pp. 420–424, 2015.

[12] J. Rifkin, “The third industrial revolution,” *Eng. Technol.*, vol. 3, no. 7, p. 26, 2008.

[13] T. J. Bergin, “A history of the history of programming languages,” *Commun. ACM*, vol. 50, no. 5, pp. 69–74, 2007.

[14] P. Naur and B. Randell, “Software Engineering: Report of a conference sponsored by the NATO Science Committee,” *NATO Softw. Eng. Conf.*, no. October 1968, p. 231, 1968.

[15] M. Shaw and D. Garlan, *Software Architecture: Perspectives on an Emerging Discipline*, vol. 123. Prentice Hall, Upper Saddle River, 1996.

[16] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach, Third edition*, vol. 1542, no. 9. Pearson International, 2014.

[17] U. W. Eisenecker, “Ai: The tumultuous history of the search for artificial intelligence,” *AI Commun.*, vol. 8, no. 1, pp. 45–47, 1995.

[18] H. a. Simon and a. Newell, “Heuristic Problem Solving: The Next Advance in Operations Research,” *Oper. Res.*, vol. 6, no. 1, pp. 1–10, 1958.

[19] J. Hendler, “Avoiding another AI winter,” *IEEE Intelligent Systems*, vol. 23, no. 2, pp. 2–4, Mar-2008.

[20] G. E. Moore, “Understanding Moore’s Law: Four Decades of Innovation,” *Underst. Moore’s Law Four Decad. Innov.*, no. Figure 2, pp. 67–84, 2006.

[21] A. Aizcorbe and S. Kortum, “Moore’s law and the semiconductor industry: A vintage model,” *Scand. J. Econ.*, vol. 107, no. 4, pp. 603–630, 2005.

[22] G. E. Moore, “Cramming more components onto integrated circuits,” *Proc. IEEE*, vol. 86, no. 1, pp. 82–85, Jan. 1998.

[23] Top500, “TOP500 Supercomputing Sites,” *http://www.top500.org/*. 2010.

[24] E. Strohmaier, H. W. Meuer, J. Dongarra, and H. D. Simon, “The TOP500 List and Progress in High-Performance Computing,” *Computer (Long. Beach. Calif).*, vol. 48, no. 11, pp. 42–49, Nov. 2015.

[25] R. F. Service, “HIGH-PERFORMANCE COMPUTING. China overtakes U.S. supercomputing lead.,” *Science*, vol. 352, no. 6293, pp. 1500–1, Jun. 2016.

[26] R. Kurzweil, “The Singularity Is Near: When Humans Transcend Biology,” *Book*, vol. 2011. p. 652, 2005.
[27] E. Knill, “Quantum computing,” *Nature*, vol. 463, no. January, pp. 441–443, 2010.
[28] P. Kaye, R. Laflamme, M. Mosca, and N. S. Yanofsky, *An Introduction to Quantum Computing*, no. March. 2007.
[29] T. Menzies, “21st-Century AI: Proud, Not Smug,” *IEEE Intelligent Systems*, vol. 18, no. 3. pp. 18–24, May-2003.
[30] M. Campbell, Hoane, and F. Hsu, “Deep Blue,” *Artif. Intell.*, vol. 134, no. 1–2, pp. 57–83, 2002.
[31] D. a Ferrucci, “Introduction to ‘This is Watson,’” *IBM J. Res. Dev.*, vol. 56, no. 3.4, p. 1:1-1:15, 2012.
[32] J. X. Chen, “The Evolution of Computing: AlphaGo,” *Computing in Science and Engineering*, vol. 18, no. 4. IEEE Computer Society, pp. 4–7, 01-Jul-2016.
[33] A. M. Turing, “Computing machinery and intelligence,” in *Parsing the Turing Test: Philosophical and Methodological Issues in the Quest for the Thinking Computer*, Springer Netherlands, 2009, pp. 23–65.
[34] J. H. Söraker and E. Steinhart, *Singularity Hypotheses*. 2012.
[35] S. Poslad, *Ubiquitous Computing: Smart Devices, Environments and Interactions*. John Wiley and Sons, 2009.
[36] A. Greenfield, “Everyware: The Dawning Age of Ubiquitous Computing,” *Portalacmorg*, vol. 0, p. 272, 2006.
[37] M. E. Porter and J. E. Heppelmann, “How Smart, Connected Products Are Transforming Companies.,” *Harv. Bus. Rev.*, vol. 93, no. 10, pp. 96–114, 2015.
[38] M. E. Porter and J. E. Heppelmann, “How smart, connected products are transforming competition,” *Harvard Business Review*, no. November 2014. Harvard Business School Publishing, 2014.
[39] International Telecommunication Union, “Overview of the Internet of things,” *Ser. Y Glob. Inf. infrastructure, internet Protoc. Asp. next-generation networks - Fram. Funct. Archit. Model.*, p. 22, 2012.
[40] J. A. Stankovic, “Research Directions for the Internet of Things,” *Internet Things Journal, IEEE*, vol. 1, no. 1, pp. 3–9, 2014.
[41] H. Sundmaeker, P. Guillemin, and P. Friess, *Vision and challenges for realising the Internet of Things*, no. March. 2010.
[42] N. Marz and J. Warren, *Big Data - Principles and best practices of scalable realtime data systems*. Shelter Island: Manning Publications Co., 2015.
[43] N. C. L. Hess, D. J. Carlson, J. D. Inder, E. Jesulola, J. R. Mcfarlane, and N. A. Smart, *Clinically meaningful blood pressure reductions with low intensity isometric handgrip exercise. A randomized trial*, vol. 65, no. 3. 2016.
[44] Klaus Schwab, *The Fourth Industrial Revolution*. 2016.
[45] H. (Deutsche P. A. Henning, Kagermann(National Academy of Science and Engineering). Wolfgang, Wahlster (German Research Center for Artificial Intelligence). Johannes, “Recommendations for implementing the strategic initiative INDUSTRIE 4.0,” *Final Rep. Ind. 4.0 WG*, no. April, p. 82, 2013.
[46] European Parliamentary Research Service, “Industry 4.0. Digitalisation for productivity and growth,” no. September, p. 10, 2015.
[47] Deloitte, “Industry 4.0. Challenges and solutions for the digital transformation and use

of exponential technologies," *Deloitte*, pp. 1–30, 2015.
[48] S. K. Khaitan and J. D. McCalley, "Design techniques and applications of cyberphysical systems: A survey," *IEEE Syst. J.*, vol. 9, no. 2, pp. 350–365, Jun. 2015.
[49] M. Jamshidi, *System of Systems Engineering: Innovations for the 21st Century*. John Wiley and Sons, 2008.
[50] J. Dahmann, G. Revobich, and J. A. Lane, "Systems engineering for capabilities," *CrossTalk - J. Def. Softw. Eng.*, vol. 21, pp. 4–9, 2008.
[51] J. Dahmann, J. Lane, G. Rebovich, and K. J. Baldwin, "A model of systems engineering in a system of systems context," *… Syst. Eng. …*, 2008.
[52] M. W. Maier, "Architecting Principles for Systems-of-Systems," *Syst. Eng.*, vol. 1, no. 4, pp. 267–284, 1998.
[53] L. Von Bertalanffy, "General System Theory," *Georg. Braziller New York*, vol. 1, p. 289, 1968.
[54] C. B. Nielsen, P. G. Larsen, J. Fitzgerald, J. Woodcock, and J. Peleska, "Systems of Systems Engineering: Basic Concepts, Model-Based Techniques, and Research Directions," *ACM Comput. Surv.*, vol. 48, no. 2, pp. 1–41, Sep. 2015.
[55] M. Henshaw *et al.*, "The Systems of Systems Engineering Strategic Research Agenda Systems of Systems Engineering," *8th Int. Conf. Syst. Syst. Eng. Maui, Hawaii, USA - June 2-6, 2013*, no. 2, pp. 99–104, 2013.
[56] A. Kassahun, R. J. M. Hartog, and B. Tekinerdogan, "Realizing chain-wide transparency in meat supply chains based on global standards and a reference architecture," *Comput. Electron. Agric.*, vol. 123, pp. 275–291, Apr. 2016.
[57] Supply Chain Council, "Supply Chain Operations Reference (SCOR ®) model," *Design*, vol. Version 10, pp. 0–23, 2010.
[58] B. Tekinerdogan and H. Sozer, "Chapter 4 – An Architecture Viewpoint for Modeling Dynamically Configurable Software Systems," in *Managing Trade-Offs in Adaptable Software Architectures*, 2017, pp. 79–97.
[59] L. Northrop, R. P. Gabriel, D. C. Schmidt, and K. Sullivan, "Ultra-large-scale systems," *Companion to 21st ACM SIGPLAN Conf. Object-oriented Program. Syst. Lang. Appl. - OOPSLA '06*, p. 150, 2006.
[60] P. Bourque and R. E. Fairley, *Guide to the Software Engineering - Body of Knowledge.* 2014.
[61] B. Boehm and R. Turner, *Balancing Agility and Discipline: A Guide for the Perplexed*, vol. 22. 2003.
[62] L. Bass, I. Weber, and L. Zhu, *Devops: A Software Architect's Perspective*. New York: Addison-Wesley, 2016.
[63] G. Piccoli, *Information Systems for Managers: Text and Cases*. West Sussex: Wiley, 2014.
[64] K. Manikas and K. M. Hansen, "Software ecosystems-A systematic literature review," *Journal of Systems and Software*, vol. 86, no. 5. pp. 1294–1306, May-2013.
[65] E. Arkin, B. Tekinerdogan, and K. M. İmre, "Systematic approach for deriving feasible mappings of parallel algorithms to parallel computing platforms," *Concurr. Comput. Pract. Exp.*, p. n/a-n/a, 2016.
[66] B. Tekinerdogan, N. Ali, J. Grundy, I. Mistrik, and R. Soley, "Chapter 1 - Quality concerns in large-scale and complex software-intensive systems," in *Software Quality Assurance*, I. Mistrik, R. Soley, N. Ali, J. Grundy, and B. Tekinerdogan, Eds. Morgan Kaufmann, 2016, pp. 1–17.
[67] V. Grimm and S. F. Railsback, "Designing, formulating, and communicating agent-based

models," in *Agent-Based Models of Geographical Systems*, Springer Netherlands, 2012, pp. 361–377.
[68] D. . Meadows, D. Meadows, J. Randers, and W. Behrens, *The Limits to Growth*, vol. 205. 1972.
[69] G. Hardin, "The tragedy of the commons.," *Science*, vol. 162, no. 3859, pp. 1243–8, 1968.
[70] P. R. Ehrlich and A. H. Ehrlich, "The Population Bomb Revisited," *Electron. J. Sustain. Dev.*, vol. 1, no. 3, pp. 63–71, 2009.
[71] J. Melorose, R. Perroy, and S. Careas, "World population prospects," American Association for the Advancement of Science, Jul. 2015.

Prof.dr. Bedir Tekinerdogan

'Since the last decades, information technology is progressing exponentially and has a pervasive and disruptive impact on the global society. From the history, we have observed that advances in technology and engineering have been critical for avoiding the Malthusian catastrophe. Currently we are facing a similar situation with problems related to overpopulation, demographic change, food safety and food security, and resource and energy usage. To effectively tackle these global societal challenges and to avoid a Neo-Malthusian catastrophe, a socio-technical, system-of-systems engineering approach is needed. Will we once again win the race against the clock?'

ISBN 978-94-6343-049-4