Digitalization and Society

© Dr. Viktor Winschel,

Mannheim, July 4th, 2019

What is the meaning of digitalization? By digitalization we mean the influence of computers on our economy and society. We explain this phenomenon by considering the similarities and differences of mechanics, computer science, and economic and social theories. The hope is to understand what we can expect from digitalization and how we can shape this process.

What is the difference between a society and a simple, purely mechanical machine? The laws of a society are created in the society. Society is changing itself and its laws and its dynamics. This is not the case in simple machines. Here the laws are given and neither the machines themselves nor humans can change the underlying laws, but only discover, describe and use them. Computer science is not so much about computers as it is about software. Software consists of languages that describe computations. Computer science is about a metamathematics, which describes the mathematical process of computation itself.

Computer science thus aims at a second level, at the level of the description of the actual object of interest - the computation. In societies, theories of society describe and change the subject of theory - the society. This phenomenon seems to be a bit strange at first. However, if we look at examples of it, we can see familiar patterns: the system changes itself when we think about it and act accordingly. In the financial markets, for example, expectation formation processes and thus theory formation are decisive. Whether a bank is liquid or not is not crucial in triggering a bank run. The mere expectation that the bank will become illiquid and thus bankrupt is sufficient to make it illiquid after the bank run - a typical example of a self-fulfilling prophecy. A finance minister who announces a tax change in the future immediately causes behavioral changes - again the description changes the system. With regard to the suitability of groups of countries to form a monetary union, it must be borne in mind that the suitability may change as a result of the introduction of the monetary union itself. Theory and practice work in loops to form and describe reality. This obvious connection brings with it massive mathematical problems that economics has not yet resolved.

This phenomenon was described by the sociologist Niklas Luhmann as the re-entry of the meta- level on the subject level, i.e. when Anna - the subject - calls herself “Anna” - a description on the meta-level. This re-entry creates the identity of the system, here Anna, and makes the difference between the system and its environment. At work is an entanglement of the meta- and subject- level. This interaction of theory or more generally of the meta-level of a description and the subject level of the description is a characteristic of living beings with cognitive abilities and consciousness and their societies. Consciousness is therefore a theory about oneself - the self- description.

In economics, the meta-level was addressed in rational expectations, although not fully mathematically captured in the depth of its problem. There is also a meta-level at work: expectations, i.e. the future, influence the present as anticipation. Tomorrow is the context of today and today is the content of tomorrow and the context of yesterday. The causality from the

future was suspect to physicists since Newton, although this causality, the teleological causality, was classified by Aristotle as one of the four types of causality. The lawyers, on the other hand, are experienced in performing a teleological analysis of a contract in order to take into account what was actually meant. This does not necessarily have to be the written content of the contract, but rather the goal or intention for the future is decisive in assessing the situation today, as a teleological interpretation.

The core of the revolution of rational expectations was the insight into the necessity and attempt to equip economically acting people in theory with the same abilities of observing economists to form and verify theories. An economist must therefore model himself. An abstract version of this phenomenon is that the object of economics can become the methodology if the economist who usually examines productions examines the production of economics itself. How do we produce economic insights and economic knowledge? The method, the meta-level of economics, is thus an economics of economics, the theory of the production function of economics itself, and thus again economics itself. This twist is important for theory building and the implementation of information and social structures.

However, we need more insights into the structure of this phenomenon. To do this, we look at a better understood field with this structure and import the insights from there into the less structured science of societies. This mathematically better structured field is computer science, which moreover also contributes to the implementation of the information structures of the economy.

The mathematics of economics used so far is not sufficient to symbolically represent the reflexive or self-referential structure as in computer science in order to understand and implement it. Technically speaking we run into the following paradox: if in mathematics everything is represented as a set, as a container with elements - as in the mathematics based on set theory and the one used exclusively in economics up to now - then the economist who is a set and is contained in the economy - which itself is a set - and who thinks in himself, about the economy - as an expectation formulated in set theory, for example, is ultimately a set which contains itself. The cat is biting its tail here. Formulated in logical terms and concepts, it means that the concepts of theory and model are much more complicated in economics, where even theory and model are negligently used as interchangeable concepts. In fact, theory and model interact. The theory is an independent object within the modeled. This phenomenon of the meta-levels is also substantial for computer science: logic as the metamathematics of proving the existence and uniqueness of the objects of mathematics became in computer science its own subject - that of computation, which from a mathematical point of view is ultimately no more than a proof or a computation and generation of a witness of the thing whose existence is to be proved.

In mathematics, the discovery of a set that contains itself led to a serious crisis at the beginning of the last century. Such not well-founded sets, which contain themselves, were excluded from the usual mathematics for lack of an apparent application at that time. For economics, this means that the economist does not obviously exist in a well-founded set based mathematics, at least not in an economy which contains him, which he examines and which he tries to influence. The modern approach to these extremely helpful meta structures in mathematics is to see the function as a basic building block and thus to build the toolbox of mathematics a level above the objects, on the functional and relational level. Objects are represented as functions. Technically, we're switching to process mode. We did that for economics.

The strange not well-founded sets, which contain themselves, found meaningful applications in computer science in the eighties: infinite data structures and object descriptions are functionally very elegant formulatable in the infinite regress of the Russell paradox as a set that contains itself. However, the history of the commonality of the social sciences and computer science actually begins even earlier. More precisely, since the beginning of computer science, if one accepts the beginning as a social scientific metaphor. This beginning can be dated to the publication of Alan Turing, in which he provided a definition of computation or algorithm that is still valid today. His approach was as simple as it was ingenious: he, a mathematician, described mathematically what a mathematician does. He has modeled himself - as economists should have done. A computation writes symbols and toggles between them and at some point the computation is finished and the result is available again as a symbol. It is not for nothing that this definition reminds us of the economist's need to model himself, or rather to model economization, as the subjects he observes do. Thereby an economist deals with processes, productions and other things that could be described by algorithms.

The common ground becomes more abstract but also more clearly apparent if we look at the equivalent and simultaneously discovered definition of computation by Alonzo Church. He has used the means of logic to investigate what we are dealing with when functions, i.e. algorithms that transform an input into an output, themselves get a function as input or produce a function as output. These higher order functions, like the Turing machine, were the beginning of what we now call programming languages. Whereby both types of programs, Turing machines and functions of higher order, compute the same set of functions and are therefore equally powerful as programming languages. Programming languages are therefore languages for the description of computations. They describe processes: first do A, then B, remember what came out of A and process it with the result of B. This is a process that describes productions, observations and behaviors.

So there are programs or functions that process other programs. If you took two recipes and made a book out of them, it would be a higher order recipe. Higher order functions can be interpreters, a software robot that can be run on universal Turing machines. They translate a program in a high- level language with abstract concepts, such as vectors or loops, into languages with low concepts that the computer can execute in hardware. These low languages can be only the famous zeros and ones at the very bottom. A computation can therefore itself be the input of another computation. Somewhere in between we are dealing with data, recursions and search functions. An interpreter with a program as a fixed point becomes evolution in economics, with the interpreter as a learning process and the program as a strategy. Only if we describe production processes as algorithms can we also ask ourselves as what one can describe an economics professor who produces economists who describe production processes. Can we also ask what a rule would be to find constitutions when constitutions are rules for finding good rules or laws, at the operational level of politics? For an economist, such higher order functions are very strange and alien. For a programmer who uses functional programming languages, such higher order functions are quite common and even the reason for the high productivity of functional programming. With these higher order functions, the programmer can program himself in a certain way by programming a programming robot, an interpreter - a higher order function, an algorithm that processes another algorithm and produces others. Data and programs are therefore not clearly distinguishable from each other. A program can be itself data for another program. For example, is 2+3 data, namely 5, or is 2+3 a program? On some of the currently most complex mathematical construction sites, mathematicians and computer scientists try to better

understand this unclear equality. Algorithms and higher order functions are a self-referential phenomenon and thus resemble what is needed to describe the process when Anna calls herself “Anna” or when the economist describes himself as an economic actor. We are thus dealing with the first important common pattern of computation and cognition: reflection or higher order functions.

In the social sciences even more complicated questions of higher order arise: if egoism can already be grasped by the simple mathematics of reductionism and altruism only by the complex mathematics of synthesis, the addition of relations, is it possible for economists who master only a simple mathematics to generate egoism by referring in their recommendations to the egoism of human beings? How is theory and reality intertwined here? Do you only recognize what you can see? Do you only produce what you can see? What can managers learn from this to drive the digitalization of their businesses? Did the shareholder value concept lead to the last financial crisis? Or was it the inability, together with a so far non-existent monetary theory, to distinguish between the object, money and the reference to it, the deposits arising as the creation of credit? Where and at what system level do debtors and creditors add up to zero? Is the global economy as a whole in debt? Complexity is manifested in the structure of how a system is composed.

This second problem of composing systems also had to be solved by computer science in order to produce complex software - that of compositionality. This is an ancient problem and one of the greatest difficulties that science has to offer: Non-reductionism, synthesis, entanglement or the transition from local to global structures, as studied in geometry, the science of the whole. There are many names and variations of this problem and it can even be assigned to the two language approaches of the Occident and Orient as object and process orientation. It ultimately has to do with something that Aristotle already noticed: the whole is more than the sum of its parts. But what is missing? The sum of the connections, functions, processes or relations between the components, which must also be added or composed, is missing. But the addition of relations is something you don’t learn in school and also not as an economist. So it is not surprising that the whole, the context, the environment, which encompasses the economy, is polluted, since the connection of the individual parts and households of the economy cannot be adequately added - i.e. if the interaction of the context, i.e. the environment with its content, i.e. the economy, cannot be captured by economists and their too simple mathematics.

Where does the search for the behavior of the whole emerge in computer science? Here the items are the basic operations that are present in a programming language, for example a loop that repeats a sequence of operations. These and other basic operations are assembled in the way the syntax allows. Syntax therefore describes the permitted relations between the individual operations. The formal semantics of programming languages is interested in the behavior of the whole, i.e. the software, programmed according to the permitted syntax and with the help of the basic operations of the programming language, the individual behavior of which is known. The analogy to social science becomes very clear here: If we take the behavior of the individual parts as the behavior of individuals, the syntax as the permitted and rule-compliant interactions in a society, which are described by institutions, i.e. the rules of the game, then a theoretical economist is interested in the behavior of the composite group of individuals and ultimately in the behavior of the economy or society as a whole. This can be the society as a whole, in political economics or macroeconomics or the composed sub-societies such as companies or households. We have now identified the two important phenomena in society, reflexivity and compositionality, also in computer science. Computer science is much better structured and researched and thus

we can import the knowledge gained there into economics in order to better structure and organize our economies. This is what is currently being observed as digitization: it is the computer scientists and not so much the economists who are changing the economy. But one should combine both points of view and this article should help.

In order to better recognize, analyze and design complex, composite structures of higher order in politics, the structures should be similar at the various levels: a function that processes a function should again be a function and not an element, function, derivation rule or integral, as in school or the economists mathematics, which is too simple and therefore often too complicated. Abstraction is a tool for recognizing similarity and patterns, and once you have found the right level of abstraction, the objects of interest should appear simpler and more similar. The pattern, as the quintessence of simplification and understanding, leads to the progress of mankind because complicated operations can be made simple. Institutions, the rules of higher order in the context, should simplify the daily human operations in the content. Higher order functions are thus a method to recognize the composite fractality of the universe and to simplify design. In our research we have sought the fractality of the economy and defined it as open games that are functions of higher order. An open game is a single decision or interaction of individual decisions that pursue their own goals, mediated by interaction with the context. Open games are formulated in relation to a context, which in turn can be nobody, another player or again a composite game. Thus an open game is a kind of fractal building block of decisions. Everything in our approach is a game, also a game about a game and a game composed of other games, is also a game again. Children play such higher order games every time they play out by scissors-stone paper, whether they will play chess or football. Lawyers call games contracts, i.e. interactions of goals, decisions and joint results. Our open games mediate between yesterday and tomorrow and thus also allow to represent and solve temporal, i.e. sequential and parallel aspects of the money loop of our money theory, i.e. the information flows within time and between investment and later consumption. Systems of systems are therefore reflexive and compositional.

In institutional economics, the question arises as to how institutions, rules or context interact with the content, how the context arises from the content and how the context constrains the content. This research direction began with the question of why there are companies and markets and in which situations we should better organize ourselves centrally, i.e. in companies or decentralized, by mediating prices on markets. The answer to this simple but complex question was again as simple as it was ingenious and far-reaching: companies are created to save transaction costs. Transaction costs are the effort required to negotiate every day all contracts with production suppliers and employees to do the business of the day. To avoid these transaction costs, people form groups called companies, rules, processes or institutions and enter into long-term contracts and relationships. It is now the case that computer science and computers have massively reduced these costs. Transaction costs include, for example, communication and search costs. The effect of reducing this important economic parameter, or context, is that there are more short-term projects organized outside of long lasting companies. So we see more markets, freelancers and markets for markets that can also be called platform economies. Some of the freelancers are happy because they are more flexible, some are not satisfied because their work is less secure. For their part, the social insurers do not always understand the effect of the process of reducing transaction costs, and above all whether they are dealing with employees or self-employed project workers, and so the bureaucracy sometimes reacts with preventive rather than enabling means.

In the political dimension, the reduction of transaction costs allows a movement towards political representation of voters at a lower level, and thus more and more elements of direct democracy are discussed. One can expect that the digitization of politics will raise the question of the nation- state itself and that we will move towards smaller and smarter city states. Here again one sees the general pattern of social systems: they change out of themselves because a parameter, the context and the content, the transaction costs, change. Algorithmization or digitization thus poses and enables new questions and answers to the question of what organizations and institutions are, how they are to be organized and what is the optimal level of control. The context interacts with the content - in this pattern we look for optimal control levels - in the symbol of yin and yang, the snake that eats itself up, the Moebius ribbon or the bottle of Klein. The interface between the context and the content we call system, boundary, organization and solutions can be inside the system or outside, changing the system.

In quantum physics it became spooky when it was discovered that an entanglement of electrons is possible over long distances. The deep causes of such a contextuality and non-locality are not understood and are still spooky. There seems to exist another global level beyond the three, four or how many known dimensions of space and time. What is understood is that two quantum systems can no longer be described separately once they have interacted with each other. It is indeed reassuring, however, that the universe at the most elementary level seems to be a relational and thus a social phenomenon. Formulated in the old mathematics of the so-called Hilbert spaces, quantum physics seems strange, but not in the newest and relational mathematics of category theory, which we now also use for our economics based on open systems. From a social science perspective, the quantum phenomenon is not unusual. Two people who meet are no longer the same, they can no longer be described individually, they have a relationship. Or to put it differently: if the observer changes the system and the system changes the observer, then we are in the middle of the mess of the social sciences - we are dealing with a new composite system. If that seems spooky, remember your mother-in-law, your better half or your tick. These encounters don't leave you unchanged.

Based on these considerations, we develop a mathematical language based on computer science and quantum physics, i.e. the programming language for quantum computers, in which we define the building block of decisions and operations necessary for economics, from composing larger decisions up to institutions. We want to be able to formulate all previous existing economic theories in this language and must now find out whether the new possibilities can solve old theoretical problems. To this end, we have initiated the next phase of our work: the ultimate entanglement of theory with the system, i.e. with the economy itself. We have therefore created a company called OiCOS, the Greek word from which the words ecology and economy, context and content, arose, at a time when context and content appeared in harmony.

The OiCOS product is a money accounting system. This can be thought of as double-entry bookkeeping 2.0. Our money accounting includes the creation and booking of central banks, money, loans, currencies and the bank of central banks. The basic Smart Contract, an open game, is the change, the tally and the blockchain of the Romans. These are contracted between strategic agents, whose interaction is in turn formulated as our open games. These agents can be households, companies, banks, governments, central banks or the bank of the central bank. These games take place in the context of rules that arise from the process of Mechanism Design, and this process can be thought of as the inversion of open games. Games produce coordinations from rules while mechanism design produces rules from the desired coordinations. If a game is based

on these rules, the desired co-ordinations should result. The technology of OiCOS is a visual, compositional and reflective programming language, as already established in quantum physics and quantum linguistics, with the aim to design, manage and understand contracts, decisions, institutions and organizational forms. Ultimately, compositionality and higher order functions promise to provide the OiCOS platform that enables the definition, exchange, and reuse of proven organizational forms. This can be seen as the digitalisation of the consulting industry or as the opening of proprietary organizational systems, i.e. Enterprise Resource Planning (ERP) or Business Process Modelling (BPM) systems such as SAP. Our language simplifies the programming of complex economic simulations and the concrete implementation of strategic interactions, the control of energy networks or other social systems. We can think about the robots for McKinsey, BCG or Roland Berger, formulated with the means of formal semantics of programming languages which studies interpreters or robots of the software industry. We can thus think of an organization that represents a process that changes processes - i.e. the reorganization that represents a reform, the constant permanent state of the process that changes processes. So it's about the proverbial institutionalized revolution - a higher order organization - that's what OiCOS is supposed to be.