



HAL
open science

The Code of Banking: Software as the Digitalization of German Savings Banks

Martin Schmitt

► **To cite this version:**

Martin Schmitt. The Code of Banking: Software as the Digitalization of German Savings Banks. IFIP International Conference on the History of Computing (HC), May 2016, Brooklyn, NY, United States. pp.141-164, 10.1007/978-3-319-49463-0_10 . hal-01620141

HAL Id: hal-01620141

<https://inria.hal.science/hal-01620141v1>

Submitted on 20 Oct 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

The Code of Banking: Software as the Digitalization of German Savings Banks

Martin Schmitt

Centre for Contemporary History Potsdam, Germany
schmitt@zzf-potsdam.de

Abstract. To the present day the history of banking software is nearly untold. While there is already some literature on the use of computers in the banking industry, most of it focuses only on the hardware and its restrictions (cf. Cortada 2006). The logic behind these machines remains untold. With the advent of the computer as a universal machine since the 1950s, business processes have been written into code, not hard wired into the machine. Furthermore, not the processor but the system software steered what was presented on the screen to the banking employee. Hardware got more and more exchangeable, while the real guiding principles of computing in action are to be found in software. This article analyzes how German savings banks used software to digitalize their business during the period of the Cold War.

Keywords: History of Computing, Software Studies, Digitalization, Financial Technology, Economic History, Savings Banks, Germany, GDR

1 Software and the Contemporary History of Computing

"But the computer is an all-purpose machine, and the computer display - a screen programmed to present text and pictures somehow stored in the computer - is a universal miraculous communication tool", Ted Nelson. [1]

"It is the history of computer software and not of the computer itself that is at the heart of the larger story of the great computer revolution of the mid- to late twentieth century" emphasizes the historian Nathan Ensmenger in his recent study about programmers and computer users in the Digital Age. [2] Especially in the banking sector the importance of software in the digital (r)evolution¹ is striking, but nearly untold. Banks were not only early user of computers, but also of software services as the implementation of computers into their business processes. Through the lens of Software, this article aims to show how the liaison of computing and a financially

¹ The r in brackets is hinting to the double nature of computerization being perceived as something revolutionary – especially in the press – while the actual change on the small scale is far more evolutionary.

powerful industry created a vast potential for change in the way banking was conducted in the 1960s and 70s, as well as the pitfalls of this development. With the example of East and West German savings banks, which had the largest customer-base of all banks throughout Europe, I am stressing the question: How was “the bank” represented in its software? Who wrote the code of banking, the programs, and the documentations? Following these central questions, it is interesting if there had been a specific socialist or capitalist way of deploying computers within banks and if one can distinguish between a capitalist and a socialist (banking) software. The comparison of the software uses and ways of production in East and West Germany presents a unique insight. Little is known, for instance, about the extent to which Eastern German banking officials had access to the technological and software developments made in West Germany. In fact, West German software products were copied and adapted to the needs of socialist banking. In this paper I'm showing in which way international knowledge transfer shaped the way banks were digitalized. It is critical how comparable institutions and their respective employees in different economic systems dealt with the new ways of intra-active data procession and the pitfalls of software production. That would have had an impact on how banking was represented inside the machine. After a short summary on what software actually was and who produced it for the banks, three use cases follow demonstrating the influence of software on the whole banking process.

The aim is not to replace a hardware perspective by looking only at software, but rather to place it in a Deleuzian assemblage. This assemblage contains technology, its users, the code and its execution in a broader understanding of the term software. Following Ensmenger and Haigh, the question how to define “the computer” in respect to software is striking. Since the 1970s there was a long debate about what a computer actually is. Especially the touring completeness of the computer, the question, what “stored-program” and “digital” meant for whom and the particular national interests made it so hard to decide. [3] A touring complete machine is a machine that can solve every solvable program given infinite time. That constitutes the universal character of every computer to possibly simulate every other machine of the past, present and future. Among others, this is the key feature of a computer that constitutes it as a general-purpose machine. The German media theorist Friedrich Kittler emphasized this forcefully in his media archaeology by defining the computer only by his programmability. [4] Not until the act of running a program, the general-purpose machine becomes specified, concrete and useful for the computer user. Thereby, it does not matter if the computer is analog, digital or even virtual. The software has the power to transform a super-fast calculating machine into every desired machine, be it a text automaton, a business machine, a passbook or a drawing device. It hints to the question how a computer was actually used by the historical actors. At the end of the 1980s, the founder of historical software studies, Michael S. Mahoney, already pointed out that a computer without software is as useless as a car

without gasoline. [5] Software had become the *intra-face*² between the computer and society in the course of computerization. Computerization I understand as the fundamental process of the widespread diffusion of digital information and communication technology and its penetration of nearly every aspect of life during the second half of the 20th Century. [7,8] In contrast to this definition I understand digitalization as the electronic representation of objects in the binary code, even though both terms are frequently used as synonyms nowadays. [9] Only through software the computer became the primary window to understand the world, the means for nearly every human to perceive, interpret and to express the world. Through software, the computer became the embodiment of the highest political, cultural and economic aspirations, as well as of regular daily business. And so Ensmenger concludes:

“when people talk about the larger process of the computerization of modern society, or speak of the computer revolution transforming the ways in which they work, live, consume, recreate, and engage in social and personal relationships, they are really talking about the history of software”. [2, p. 6]

But thinking of software as a product is misleading for analyzing the process of computerization. Software was far more than just an executable program that you buy on a data carrier or download like you purchase Microsoft Word for having a customizable word processor.³ Therefore a conceptual history is necessary to understand the different layers of meaning that are encapsulated in the term software. At the same time, software history already tells a lot about the recent historical developments in a time of ongoing commodification.

1.1 A Conceptual History of Software

In the late 1950s and the 1960s, computer people – experts like programmers, system analysts, technicians or administrators – understood software in two different ways. On the one hand they understood it in a narrow sense describing programs to develop other programs and to control the hardware. On the other hand they understood it mainly as a strand of systems, services and support. [10] Programs were seen as universal mathematical solutions that nobody could own. Consequently, programs had no name. Software in the early period of computerization combined whole systems of machines, their instructions, the organization, programmers, users and processes. Therefore, software can be methodologically described as a Deleuzian assemblage combining independent elements within a temporal liaison. The term software was first mentioned by the statistician John Tukey in 1958 as everything beyond the tubes, transistors and cables. The Oxford English Dictionary is enlisting the first use of the

² The term *intraface* is used here analog to the term *intra-action*. It should point to the understanding in the Latour sense that the division of men and machine or in a higher sense between men and nature is non-existent. Men are always part of their environment and therefore are *intra-acting* with it. Cf. [6]

³ Even though buying something on a data carrier already is outdated, too.

term for the year 1960 in the journal *Communication of the ACM*, whereby the author is describing COBOL as “software”. [11] Its semantic change from a narrow to a broader, system oriented meaning bears to the height of system theory in the late 1960s. To give a concrete example: If a savings bank wanted to digitalize its transactions, the bank not only had to write millions of lines of code or to buy a fancy product. It had to analyze its business processes up to the single steps of action and had to fit the computer into it – or adapt the processes to the machine. The bank had to reorganize its personnel and its organizational structure, it had to train the users, buy the peripherals and write new instructions, manuals and documentations. In a nutshell: the result of the computerization was software.

For a bank, software in the 1950s and 1960 was more like hiring a consultancy than buying a product. [12] But the consultancies of choice in most of the cases were the huge hardware producers who offered software as service together with their machines. Sales persons at IBM not only needed to understand the machines they were selling. They also had to analyze and understand the way their customers worked and had to offer solutions of how to optimize it. Seldom was this an easy task as it meant change to established hierarchies. Therefore, it is important to tell the story of the computerization of Germany not only as a success story of rapid diffusion of a superior technology. It was a conflictive process, in which software can be described as a Deleuzian assemblage per excellence. Its heterogeneous parts were fighting for power, sovereignty and authority. [2, p. 7] Whilst Nathan Ensmenger and Rob Kling are focusing on the new power of the programmer and the IT-departments within organizations, the historian of technology David Gugerli argues that standardization and strict rules of usage, language and education limited the creative freedom of the programmers severely. [8], [13]

In this respect it is striking to look at the computerization of the former GDR and their socialist rule of language with a wider understanding of the term software. Computer people in the GDR were talking not about software but about “system documents” (“Systemunterlagen”) for a long time. The different denomination corresponded with the maxim of the party elite to use the German terms rather than the English ones in a battle against a perceived imperialist American influence through technology and language.⁴ But at the same time they also distinguished themselves from the Russian usage of *matematicheskoe obespechenie* for Software, which meant as much as mathematical supply.⁵ *Systemunterlagen* catches the initial understanding of computer usage in the 1960s much better than the term software, which is buried under the different layers of meaning that software carries today and that is bound to a binary separation of hard and soft. *System* not only meant the actual computer. Rather it described in a cybernetic way the whole environment of the computer, for example the division that ran it.

⁴ Up to the 1980s, “Computer” in the GDR were called “Rechner” or “elektronische Rechentechnik”, “Hardware” was called “Gerätetechnik”. On socialist language cf. [14]

⁵ Thank you Ksenia Tatarchenko for that hint.

Not until the middle of the 1970s did the term “Soft-Ware” become established in the GDR written with a hyphen analog to the term “on-line”. According to Simon Donig this hints to the fact that completely new concepts like software as a product, the microchip or the wafer had been adapted by the engineers, whilst they kept well-established concepts like tubes (“Röhren”), interface (“Schnittstelle”) or input-output device (Eingabe- /Ausgabegerät) in their mother language. [15] And even in 1981, the lemma “software” in the encyclopedia of economy of the GDR simply linked to “Systemunterlagen” which shows the persistence in the socialist use of language. [16] This is also probably due to the relatively strict embargos for high technology during the height of the Cold War that forced the GDR to self-production and also prevented a language import.

The hyphen in the later used “soft-ware” shows a development that began in the United States in the late 1960s. The media theorist Wendy Chun described it in her book “Programmed Visions” as software becoming a thing, a “ware”. Beforehand, software was rather a service for the specific use of a computer within the company priced in labor cost per instruction. With the ongoing pervasion of the computer in business in the late 1960s a shift of meaning took place from “programing” as a verb to the noun “program” as a category. Software more and more became a product with a fixed price and branded, an all-in-one solution that was patentable, copy-able and standardized to be used in various different contexts without any big change. [17] On the contrary, in the GDR not only the way of thinking about software but also the bundling remained mainly unchanged until the mid-1980s. Even though Europe didn’t follow the US-example of patentability, the understanding of software as a product diffused with the data carriers that fed the demanding drives of the global PCs. It is a historical irony that while on a global scale the materialism experienced its ultimate failure with the end of the Soviet Union and the fall of the wall, it stroke back in the sphere of the digital. [18] While Reagan stated at the Lomonosov University in Moscow 1988 in respect to digitalization that “we’re breaking through the material conditions of existence to a world where man creates his own destiny” [19] these material conditions were set in the background. Judges in the US covering the question of software patentability in 1994 drew on the fact that the programs changed the material state of the computer in a temporary memory dump. This materialization of thought was then treated in the same way as every other material product. [20] In the mid-1990s the transformation from software as a service to software as a product was accomplished by the interest of a grown up software industry.

Despite the semantic closure for the protagonists the term software somehow stayed hard to grasp. Michael S. Mahoney explains that with a hint to the active character of software in action. For him, software is “elusively intangible. In essence, it is the behaviour of the machines when running. It is what converts their architecture to action, and it is constructed with action in mind; the programmer aims to make something happen”. [21] This purposeful component of software as active change makes it the perfect category for analyzing the computerization of German banks in a historical process – a process that is deeply coined by intra-action between technology and society. As a running computer never could be described in total, software also

stayed something ephemeral, intricate, it was the chaotic and rhizomatic. [22] During the process of computerization, hardware supposedly only got faster, smaller and cheaper. Moore's Law could be interpreted as the technological embodiment of the belief of progress that was so symbolic for the post war period up to the mid-1970s. It was the pendant to the West German "Law for the advancement and stability of growth in the economy"⁶ of 1967 or the belief in the unstoppable progress of socialism in the East. By contrast software often reflected the dirty reality, broken dreams and stony paths of implementation. Shortly after the first commercial computers were sold, experts drew on the undersupply of usable applications for them. Thereafter in the 1960s the costs for the programming rose and the software grew more complex and unreliable. For Nathan Ensmenger the discussion of software crisis within a larger discourse of computerization is an astonishing constant of the history of computing. It began with the shortage of applications in the early stage of computerization. It continued with the complaint about the perceived lack of programmers, the unreliability of the solutions and the exploding costs of software production. The final reverberation of this is the outsourcing of programmers to lower developed countries and the recent discussions about algorithmic produced software through "artificial" intelligence. Therefore, Ensmenger comes to the conclusion that software history is a history of failure, abortive developments and contradictions in contrast to a smooth progress of hardware. In short: Moore's Law did not apply for software. Software always had to be renegotiated. It had to be negotiated because of the unintended consequences for the companies who drove the computerization, not because it was "hard" as the famous computer scientist Donald Knuth stated. [23] This is an important point, even though Ensmenger underestimates the contradictions of hardware development. Recent studies of our research project are showing that the history of hardware and its diffusion was not at all smooth and gentle. Moore's law finally came true, but more important is the point that it inspired the way developers, programmers and users on the whole globe thought about information technology and what you could do with it – also in the banking sector.

1.2 Banking Software

Looking at the use of software in the banking industry, the thesis by Ensmenger of software as a problematic factor in history has to be slightly adjusted.⁷ The software story was by no means only one of failure, shortage and disappointment. First of all, the banks as customers demanded working software as the integration of computers from early on. Otherwise they chose another producer. Software sold hardware even if both tended to be unreliable in the beginning. Despite the delays in implementation and the fact that banks stuck to use computers in the very same way they used punch

⁶ "Gesetz zur Förderung der Stabilität und des Wachstums der Wirtschaft (StabG)".

⁷ "Compared to the history of computer hardware, which is characterized by regular and remarkable progress, the history of software is replete with tension, conflict, failure, and disillusionment" [2, p. 10].

card technology, the basic applications worked productively after a short period of adjustments. Most of the time, these applications were so successful that they ran for several decades and on different computer generations. Often, on a new computer generation an emulated version of the old software ran. That, in return, caused a magnitude of possible errors and problems. Path dependency was even more the case in the GDR where the savings banks had to struggle on the one hand side with scarcer resources and a lower importance in the economic system. On the other hand they had to fight with the error prone hardware and the long delays by the state-owned combine Robotron. In banking, software could also be described as the constant against a fast changing hardware – not the other way round.

Let's take a concrete example in the late 1960s to understand how computers were implemented into business processes: accounting. The primary focus of the planners in adjusting the system to the rising transaction and account numbers had not been computers. They were more interested in measuring how the current system worked, tried to understand it and to model its behaviour. Then they adapted this system to the requirements of computing. Therefore, other actors entered the stage: sales people, technicians and programmers. The initial design had then to be cast into a system architecture. This again necessitated the collaboration with several organization experts, users and technicians. Rooms had to be found and spatial ordering of people, machines and working processes had to be produced. The executives had to supervise the steps and made changes. The production of the code of banking becomes vividly graspable in the sources of the Municipal Savings Bank of Stuttgart. IBM Germany, located in the close proximity in Böblingen, not only sold them their machine. They analyzed exactly how the bank worked and optimized the processes. [24] After the programs were written using higher level programming languages or Assembler, they had to be tested and debugged. This again could change a lot of the initial plans and needed its space and machine time. In the GDR this scarce resource was highly contested so that often software was badly tested and had to be rearranged afterwards. The consecutive step then was to complete the documentation and to write a manual for the accountancy system. The employees that used the system had to be trained in how to use it what ranged from the high level executives down to the regular employees in accounting. Ian Martin has shown for Martins Bank in England, that in the beginning there was a strong continuity in employing women as machine users. They just were transferred from the machine room in the back-office and their work with the business machines to the computer centre of the bank. [25] This pattern becomes visible as well in the Savings Bank of Stuttgart, where the punch card technology in Germany was far more widespread than in England. The first jobs that became obsolete in Stuttgart were those of the punch card division, primarily a female domain. [26]

Today it somehow sounds strange but the system itself had still to be operated, as sophisticated operation systems had not been developed, yet. The operators of the savings bank had to keep the different programs in the right running order, had to check if they ran properly and had to take over or archive the results. Finally, the software environment had to be maintained and adapted to changing usages.

Altogether, this maintenance could eat up to 2/3 of the overall costs of the software. [17, p. 118] As Karl Ganzhorn, former developer at IBM Germany writes, the maintenance of the computing environment occupied so many programmers in the late 1950s that they had not much time left for new developments. [27] It is astonishing that the executive board of the savings banks on the one hand planned with the technology over long periods of time, more or less ten years in the beginning. The costs of computerization were calculated meticulously. On the other hand they seldom realized the hidden costs of software in the long run as they mainly hoped for rationalization wins. The user experiences were important for that.

In the early years of the digitalization of banking from 1954 to 1962, banking executives as well as employees first had to become familiar with the new technology. They relied heavily on the experiences they made with punch card technology and used computers as business machines. Punch card machines had been deployed in German Savings banks since the 1920s and were widespread in the post war period. But soon they had to realize that computers were more than fast calculators built of sheet metal and electron tubes. Inside the machine worked algorithms that soon became the core of banking. At the outside, working processes changed in intra-action of the computer-bank systems. For a longer period of time, digital processes didn't replace analog ones but ran together in the asynchronous tact of the workflow.

Therefore, when a bank bought a new computer not only the technical facts were important. It was the integration of the machine through software that mattered in a technical environment of growing complexity. Software was needed that kept the machines up and running effectively. So it is no surprise that in 1967 Heinrich Fuchs, head of the organizational department of the Municipal Savings Bank of Stuttgart emphasized that "the performance of this equipment depends not only on its technical specifications but mainly on the quality of its programs". [28]

After the IBM 1401 that they bought in 1960 proved to be reliable, the Savings Bank of Stuttgart decided to upgrade their machines while moving into a new building. Large scale computing always was dependent of space. It was a close run between IBM and Siemens for the contract of a new data centre for the savings bank. IBM had huge problems designing their operating systems for the system /360 but succeed in presenting a preliminary version for testing. The software of its main competitor Siemens was in delay, so this was one of the main reasons why IBM got acceptance of the tender. [28] Software sold the hardware.

One telling example for the importance of software in bank computing since the very beginning of the computerization of Germany in the 1950s is the Zuse KG and its computer Z31. While Konrad Zuse as the inventor of one of the first digital computers is already well known, the fortune of his economic venture is not. The Zuse KG was in comparison to the huge producers like BULL, IBM or Remington Rand only a midsize company. Employing 1200 employees at the very height of their success, they built specialized computers which were state of the art in research and specialized usages. The Z31 was intended by the engineers to open up a new market base in the economy. As a smaller office computer automating routine jobs it was explicitly developed for the usage in banking and savings banks. Even though the

hardware was fully developed and produced, the computer never had gone into sales because of missing software. Despite their rather sophisticated sales team and their programs to educate computer users and programmers the Zuse KG didn't have the financial resources to develop a clear strategy of how to integrate the Z31 into the business processes of the bank in advance. They lacked the programmers to write the code and missed the analysts who understood banking. [12, p. 70], [29] Because hardware producers had to provide software or massively helped their customers to integrate the computer into their business, the Z31 failed at the market despite its state of the art hardware. Probably worst of all, the Zuse KG missed the international contacts to communities of invention to fill this gap. What an advantage this could have been showed in the case of Heinz Nixdorf. With his company Nixdorf AG he was the founding father of specialized bank computing solutions. In the 1950s, Nixdorf realized a deal with EXACTA, a midsize business machine company who also had the license for BULL computers in Germany. In the end it worked out that BULL sold Nixdorf computers abroad. Nixdorf profited from the knowledge of BULL concerning the usage environment of computers. In the 1960s, Heinz Nixdorf succeeded in persuading Otto Müller, a German engineer of IBM Laboratories in Yorktown Heights to come to Paderborn. The experiences of Müller at IBM in the US helped Nixdorf to produce the System Nixdorf 820 that was a huge success at the banking computer market. [30]

Aside from the operating system the banks needed a wide range of sophisticated controller, monitoring and utility programs for running the machine. That did not necessarily mean application programs but a layer between the core of the machine and the application layer. Together with the operation system these programs built the layer or the "platform" on which the applications were based. As mentioned before, building up this platform and the applications on top of it was a huge challenge for the hardware producers. Out of their experience in the business machine market and a huge capital stock, IBM built up a specialized division for banking from early on. They provided customized solutions for their huge machines and helped massively to integrate them into the daily banking business. They gave courses and instructed the workforce and the technicians. Thousands of employees ran through those courses and their follow-ups within the companies. This was the dominant way computer knowledge spread in the early days in which *Informatik* was still in its formation phase. Often the producers even send their own employees into the banks for a period of time besides the basic technical service. Banks not only bought a machine in the 1950s. They bought a service. And this service was customized as far as possible.

A good example for that is Winfried Ferger of IBM Germany. He began there as Special Engineer in the division for customized solutions. IBM Germany was one of the most important offshoots of IBM, focusing especially on the production of mid-size hardware and the creation of software. [31] He describes a close collaboration between the customers and producers in the adaptation of international produced hardware for regional and task specific needs. In an interview he recounts:

"During all of my life I only developed things who nobody developed before. I always adapted existing things to the requirements of banks or other clients. First I

looked at the possibility for realizing a specific need of a customer. Then we developed it". [32]

Here it becomes clear that banks and especially the savings banks were the processors of the Digital Age that influenced the course of the technological evolution – also in software. Products they demanded often found widespread adoption in the whole industry and afterwards spilled over to other industries as well. Examples are encryption or data procession of information retrieval that were adapted to their needs.

1.3 The Producers of Software

Asking for the producers of banking software one encounters a multitude of different players. Not surprisingly, the regional computer centres of the savings banks played a crucial role in the digitalization of the banking system. This applies for both German states despite their different political and economic backgrounds. But the computer centres seldom produced their own programs. They rather defined the requirements for software service orders or adopted the software. They were especially important for the wide range of midsize and smaller institutions typical for a large share of savings banks. The several huge institutions in the urban centres were forerunners raising the general awareness for software solutions, but their requirements differed severely in scope and depth. Many of the programs came from large hardware producers who specialized on banking technology or from the late 1960s onwards from a growing software industry. [27, p. 51-58]

In the West, the German Savings Bank and Giro Association (DSGV) and the several federal state associations worked as distributors. They not only ordered the software solutions and made them available to the several savings banks. They also collected the experiences made by the institutions and presented them in aggregated form for example in the periodical "Betriebswirtschaftliche Blätter". They coordinated the programming effort and presented best practice solutions. Many savings banks tried to get their systems implemented nationwide through the DSGV so it was also a tool of influence and a point of conflict. Its competence and means for existence always had to be renegotiated. [33] Using a wider software term, also in-house developments like the one of the Bavarian Savings and Giro Associations are coming into view. On a modular base, they tried to develop holistic solutions for all parts of the banking business especially for midsize and smaller institutions and distributed it openly.

In the GDR, the situation of software creation only differed slightly. First of all, the banks expected the combine Robotron or its precursors to deliver the basic software bundled with their rare computers. Concerning the applications, Robotron lacked the capacities, the resources as well as the knowledge about the different contexts of usage. Therefore there was a clear separation and Robotron never produced any banking programs for the East German banks, even though they cooperated with Siemens in the 1980s to build a system for international clearing. [34] But also on the system level there was a shortage despite a sophisticated

theoretical level of the mathematical foundations of programming in science. Lacking workers in general, computer people in the GDR were a scarce resource and computer knowledge was not widespread. This was the flipside of their high expectations. In 1966, the head of state Walter Ulbricht cited publicly this shortage on both sides as he stated: “We need more programmers! But also the managers in the party, in science and economy need a better knowledge of modern computing”. [35] So it was no accident that the application field centred on programming system SOPS was already developed in late 1960s. It should provide basic general solutions for process automation in various fields like SAP did successfully with its suite in West Germany in the late 1970s. But SOPS was mainly implemented in the industry. [36] In charge for the distribution of the software solutions were the Ministry of Finance and the Central Bank. In most of the cases they ordered the software solutions at the “VEB Datenverarbeitung der Finanzorgane”, the equivalent to the computer centres of the Western German savings banks. But in comparison to the West German banks, it seems that there was a stronger knowledge transfer in the East. In the course of the mutual coordination of computer production under the COMECON ESER-program,⁸ banks of the Eastern Block exchanged their experiences on banking information technology on a regular base. They even agreed upon collaborative software creation. [37, 38]

Looking at the GDR is also telling in respect of the concrete programs used. Lacking the foreign currency to license programs from the West and understanding software not as a product, institutions of the GDR tended to copy programs from the West under different names. This matches the strategy of hardware imitation used for example in the creation of the R300 as inspired by the IBM 1400-family, or the R40 as a IBM S/360 adaption – both computers were the most common ones used in the GDR up to the 1980s. In the 1960s, Robotron and his precursors concentrated mainly on providing system software. Basic software wasn't provided and the users had to develop them all by themselves. With the rise of the Digital Age - understood as the ongoing diffusion of small and midsize computers and the increasing knowledge of computer usage in the GDR - also the expectations of the computer users rose. They demanded specialized software solutions for the conditions under socialism, namely a state oriented, planned economy. Robotron reacted late but founded the “VEB Robotron-Projekt” in Dresden in 1984 as their software house. Suffering under the ever-severe resource shortages they mainly continued the strategy of renaming foreign products and deploying them. Reacting to the demands of the users, they also tried to develop some specialized products for a socialist economy. However, in the meantime Western products reached a huge diffusion and the users rejected the incompatible solutions of the “VEB Robotron-Projekt”. In result, these projects were abandoned thereafter, despite the large investments already made and the goals that were already reached. [39]

⁸ ESER = “Einheitliches System Elektronischer Rechentechnik”, *standardized system of electronic computers*.

2 Information Systems

Banking information systems are a type of software that was developed in the late 1960s and reached a huge adoption afterwards. After defining the term information systems I am taking a closer look at the program system SODIS⁹ of the savings bank in Saarbrücken, TELDAS as the system of the Municipal Savings Bank of Stuttgart and finally at the information system of the state bank of the GDR and its conditions of international development.

As an “information system” Horst Stevenson, a widely received banking automation expert, understood in 1973 a “form of data organization [...] realized with electronic data processing systems [...] supporting certain operations and routines within information and communication processes”. [40] First of all, this means that information systems in this understanding are necessarily computer based and binary-digital. Second, their means was to support, not to substitute the employees through providing the right set of information in the right amount at the right point in time. Information systems were a reaction to the perceived masses of data produced by the machines and the growing customer base since the early 1950s. The contemporaries had difficulties to extract the relevant data out of these masses. The context of this perception in the industrialized countries are the late 1960s as a time of apparently rising complexity and failing steering mechanisms in a globalizing world. [41] It is important to keep in mind though, that in all times people had to deal with large amounts of data and the complaint of drowning in data was not a new phenomenon even then. [42, 43] On the one hand information systems were closely related or based on technological developments like real-time accounting, on-line data transfer, databases and display technology. On the other hand they represented a method of integrated process engineering. They were the product of knowledge about the usage and the integration of the computer into banking.

Stevenson understood information systems as the dawning of a new era of computing replacing the process-oriented phase of data processing. Process oriented data processing meant the division between certain areas without or with little exchange of data between divisions. Managers and other employees only hardly could interconnect the data sets they possessed. Banks had therefore to record certain data several times. The problem of double recording applied even more for the savings banks of the GDR who had to fight longer against the problems of divided process chains between digital and analog – as well between different business areas as between whole institutions not yet computerized. For a long time they kept on specializing the integration for certain areas like the generation of statistics. The output of statistics was more or less useful for managerial demands but seldom for the daily business. Savings banks began to integrate their divisions into the information systems step by step then, beginning with the most important areas. Not until the early 1980s, *total* information systems were achieved capturing all parts constituting a bank. Even then, the cybernetic dreams of universality were never quite reached. [44]

⁹ The term SODIS was an abbreviation for the German translation of real time transaction.

In a nutshell, information systems developed out of integrated data processing that was limited to a certain area. [44, p. 27]

To look at information systems with a wider definition of software makes sense out of a double reason. In the first place information systems are composed of the “total of interconnected workers, machines and organizational institutions that are data processing or informational productive in the course of the managerial process of the enterprise”. [45] Even though in this definition by Dahms and Haberlandt the data itself is hardly missing, information systems reached way further than just the lines of code it was written in or the machine it was run on. It rather described a rhizome like conglomeration that matches the initial understanding of a wider software term. In the second place, it had a clear focus on the user aspect and described rather a service or a solution to a daunting problem like software did. Besides these two points, there are some additions to be made. The term used in the 1970s was deeply shaped by cybernetic principals. It is important to keep that in mind while looking at the expectations banks projected onto the information systems but also while analyzing how these systems were marketed to them. Cybernetic models often suffered from an exaggerated universalization of their scope, especially in the case of social cybernetics where different elements were described with similar terms and afterwards appeared similar. The tendency for methodological overstretch applied for a range of areas of use when the different elements of information systems were unified under the language of information. The problem is clearly graspable in the use of the term information one can find in the sources. The idea that data got distilled to information within a socio-technological process prevailed. Information was seen more like a material resource than a constructed entity. It is important to avoid a teleological narrative of the savings banks on a path to ever-greater knowledge and control through technological advantages. Software as an ephemeral and problematic concept is helping to focus on the dirty reality of computerization. Information systems were also a factor for greater uncertainty, perceived disorder and confusing fragility. Viewed this way, the term of information systems under the auspices of software is a tool for applied data critics. [46]

2.1 Applied Information Systems: SODIS and How the Bank came into the Computer

In 1967, the Savings Bank of Saarbrücken, located in West Germany at the French boarder, proudly presented their internal software system SODIS to the public. [47] The institute, one of the most computerized institutes of the republic, decided to use the medium of the IBM bulletin to show their recent advancements. They stated that the main goals for the managers introducing this system were the rationalization and acceleration of every process inside the bank. In the mid-1960s, their account data was still recorded analog and then transferred to punch cards afterwards in the punch card division. With the help of an on-line system consisting of electronic business machines, a brand-new IBM S/370 and the data connection lines, data could be directly produced born-digital. This also meant that the account data was immediately

up to date to speed up a whole range of processes. Parts of the bank were now accessible fully digital and in real-time. The project was located at the borderline of classic data processing within the divisions and the new information systems. The head of the organizational division Ingo Holtzmann stressed this transitional character of the system that “should not be misunderstood as a sophisticated information system in the classical sense of the term. It should rather provide precise criteria for decision in the context of immediate processing of transactions”. [47, p. 44] Only in a further step it should evolve into a full-fledged management information system. As Stevenson showed, the transition from integrated data procession to information systems was fluid.

The special role of software within information systems becomes clear in the project description of Holtzmann. He begins with an extensive description of the universal character of the computer. For him, universality was rather a disadvantage because the “business of savings banks demands specialized machines for the individual ways of procession”. [47, p. 45] The solution of choice in the 1960s still was a combination of hardware and software with specialized keys as the hardwired and static interface to the machine. The advantage of a flexible defined button realized through the interface was not graspable for the contemporaries yet. But the core of the solution was software. It transformed the problematic universalist computer into a specialized and useful device inside the savings bank business. So Holtzmann concludes, “the result of all organizational planning [of the bank] in programmed form is the SODIS program system”. [47, p. 47]

It was produced through two teams of programmers that were separated into on-line recording and further procession. For the sake of interconnection, the planners insisted on “a precise definition of every process of business”. [47, p. 48] It becomes visible how the organizational department recorded every step of work inside the savings bank business, standardized it and made it machine readable to support it by information technology. In short: banking was depicted in software. Also the style of work of the programmers became part of this standardization process. The management saw their work as opaque and what they did for many spectators seemed more like a *black art* than actual work. This could cause a multitude of problems and delays as the rigid plans seldom left the freedom for the creative thinking that was necessary for a good systems design. In the end, the drive for standardization was also a way to confine the power of programmers in the process of writing the code of banking. Holtzmann: “Analysis was produced for every program. Regulations for programming were circulated internally. Detailed trainings and narrow introduction influenced positively code, text and completion of the programs”. [47, p. 48-49] Not only the software environment but also the programmer himself should deliver expectable and increasingly standardized products. For the 1960s that still experienced a huge error rate in computing, this was a huge step forward. But at the same time it was the confinement of creativity in the struggle for organizational power. In the militarized words of the management oriented “Zeitschrift für das gesamte Kreditwesen” this struggle becomes visible: “The Chief Technology Officer and his reserve unit will grow in power, but the decisions had to be left in the hands of

the management”. [48] Therefore, they also had to acquire competences in EDP to direct the change, the article concludes.

The programs of SODIS were written directly in Assembler, a machine language. In comparison to higher programming languages this had the advantage of optimized usage of computer time and a lower usage of memory. [12, p. 123] Even though the engineers realized the daunting problem of writing code directly for a specific machine and tried to get independent, their solution of standardized input and output codes was only a transitional one. Writing for a specific machine always meant higher maintenance costs and a greater effort to migrate a coded solution to a new hardware environment. [27, p. 51-52], [47, p. 49] So SODIS was a transitional solution in theory as well as in usage. But in the eyes of the management of the Savings Bank, the program proved useful to them and they expanded it to other fields and branches. At the 10th anniversary of SODIS, director Lehberg not only stressed the pioneer work his institute had carried out. He also pointed to the huge interaction between the business of banking and the system. “Besides the janitor, the new working process reached out into every division of the bank. The change was so strong that sometimes one could barely identify the old processes afterwards”. [49]

What exactly did change? First of all, the direct procession led to a massive acceleration of payments and withdrawals at the counter. While beforehand transactions could take up to 15 minutes, now a single transaction only took 15 seconds – including the printing of a receipt. [48, p. 977] The account was up to date immediately, while transactions in the old system only were processed after the cut-off time, in most Savings Banks at 4pm. The teller could see immediately the balance of the customer and if other transactions had taken place. Second, it also meant a huge acceleration in back office as the lavish sorting rounds of data sheets or punch cards became unnecessary. Third, information systems like SODIS meant a new possibility for the management to grasp the Savings Bank in the very moment of a request. The bank became visible to directors like Lehberg in (nearly) real time numbers and data streams. Or in the words of the before mentioned journal this meant “faster and further diversified balance sheets, stock overview but most importantly an overview of the cash-flow”. [48] It is striking that here even conservative business experts began to dream about the possibility of getting an “insight into structural trends on national economies” [48] given just enough data of different clients. These dreams never came true and in 1967 it still was a long way to go until the whole bank was integrated into the information systems. The trend towards a higher insight into banking can be seen in the light of the professionalization and scientization as writings of contemporary history are highlighting for the 1960s-1970s.

On the international stage, information systems like SODIS were an important topic of discussion, even though the cooperation was limited mainly on knowledge exchange. At the third international conference on automation in Vedbaek in 1967, delegates from 16 Western countries came together to discuss the recent trends in savings bank automation. Besides USA and Australia, all of the other delegates came from Europe. The German delegation showed a great interest in the question of software and mutual exchange. Director Claus of the Institute for Automation (IfA)

described their mission in producing test cases of working processes and sample programs that should become freely distributed to reduce the burden of programming and asked for similar examples. The sources are indicating that European software exchange was developing with a core in Scandinavia. [50] In contrast, the US-delegate stated that the local savings banks still acted with great caution concerning the provision of programs. The conference protocol ends with a statement of Dr Richard Nowak from the German Savings Bank and Giro Association. He concludes that “leaders should acquire a straight approach to automation; personally they might be enthusiastic about the new technology or condemn it. But they cannot escape [...] the development”. [50] It is telling that information technology at this point in time already was seen obligatory by certain banking elites – a rhetoric figure well known in the history of technology. In 1973 the International Institute of Savings Banks listed the efforts for mutual software exchange in their regular reports on international savings bank automation. For Germany, they noted that the different programs provided by the Institute for Automation were already installed in over 16 institutions. They even were willing to provide them to other European Savings Banks – even though for a fee. [51]

All in all, SODIS developed in the context of early information systems still limited to certain areas. Systems like SODIS formed the base for further development of higher integration, like the next case of TELDAS shows.

2.2 TELDAS as Consultancy: the Making of Path Dependencies

A similar development experienced the already mentioned Municipal Savings Bank of Stuttgart, another forerunner of German computerization in using IT. After moving to their new main branch on Königsstraße 3 in 1967, they now had the space for purchasing a new mainframe computer. Similar to the purchase in 1960, as they acquired an IBM 1401, they sought IBM for an exhaustive report on banking automation at their institute. IBM delivered. As they proudly pronounced, “our report based on a exhaustive study our your institute”. [24], [28] Not surprisingly, the document sounded more like consultancy report than an advertisement brochure for a computer system. The IBM consultants broke down the working processes into single steps, reordered them anew, measured the processes and compared them to the existent non-DP processes in respect to duration, efficiency and cost. While in the 1950s, the Municipal Savings Bank of Stuttgart had been a customer of the French *Compagnie des Machines Bull*, in the 1960s it had already switched to IBM machines after intense discussions. A strong argument was the close proximity of IBM Germany. They had opened their headquarters in Böblingen and not in Bonn or Berlin. IBM Germany profited from its status as a loyal customer of the savings bank who gave them credit – one example for Savings Banks as processors of computerization in Germany. But even more so IBM profited from their sophisticated sales team with expertise in banking automation.

But the report was only the beginning of the digitalization of the bank. On the two brand-new IBM Systems 360/40 their IT division ran the remote data procession

system TELDAS, a system quite similar to SODIS. In the beginning, the system was used for the handling of savings transactions. The employees could now instantly book savings transactions “on-line” into the account of the customer through terminals connected with the mainframe. They also were now in the position to directly print on the savings book in same work process. Step after step the Savings Bank of Stuttgart integrated more and more divisions and branches into TELDAS. In 1971, giro traffic was integrated into the system by the DP division. In 1973, TELDAS also covered foreign currency, mortgages, securities and standing orders. On top of this database eventually the customer information system (KIS) was installed, provided by the Institute for Automation. With the customer information system it was possible for every bank employee to get a fast overview about every customer and to individualize the services offered to him. Missing possibilities were made visible by the machine suggesting new products. The deployment of the information systems took place in the strategic reorientation of the German Savings Banks towards a customer centred strategy. [52]

Looking at this specific system installation over the period of only 6 years, software had gotten more and more important in respect to the effective functionality and it is striking how interchangeable hardware had become. TELDAS already did not run anymore on the initial IBM S360/40s but on an IBM S370/158. The hardware replacement changed not that much in respect to functionality. Banking had become code, not a machine. This is referring to “The Government Machine” by John Agar who had the thesis that the British state in the 20th Century had become a bureaucracy machine similar to the punch card systems it relied on. [53] For the banks it was the software, not the transistor and it was the integration of the data, not of the circuit that mattered.

In 1979, the Savings Bank of Stuttgart was integrated in total in TELDAS. In the annual report of the Savings Bank for the year 1979 it is simply said: “All customer related areas are fully organized within TELDAS”. [54] In the following year, all 250 branches were connected to TELDAS through remote and up to 420.000 transactions were realized through the system. But this did not mean that the bank was available to the management in total. The report of the International Institute for Savings Banks states that in Germany the high dreams of information system universality were buried underneath the details of daily business. [55] The story also is an example for how hard it was after the full integration to step outside the path dependency of own specific supplier. Even though on a higher organizational level, the banks tried not to get depended of one hardware producer, all competitors from now on had to build their systems compatible to TELDAS. Up and running, the software systems were productive for decades as they carried the code of banking. So software systems in Germany seldom changed and if they changed, then because banking itself was transformed in close intra-action to information technology.

2.3 Informational Socialism: The Information System of the Central Bank of the GDR

In comparison to the FDR, in the GDR the international community of invention was stronger in respect of information systems. During the phase of intensified computerization in the years 1968-1972 under the rule of state driven cybernetics the project “open accounting, transaction clearing and accountancy” built the base for further integration. Upon this transaction data, the information system for internal bank data should provide primary information to the leading cadres for further use. [56] My thesis is that this generally implied a better control of the planned economy through knowledge about the currency flows in a political phase of new economic concepts under Walter Ulbricht and a loosening of direct control. Flows of current and flows of currency converged in the eyes of the party elite.

The State Bank of the GDR developed the information system on a modular base that at the same time “evoked integral effects of rationalization as well as the preparation of information for field specific and management tasks”. [56] The discussion and the development were very similar to that in Western Germany, even though it took place a bit later: After the first successful projects of computerization it was now a challenge for the leading personnel not to drown in all the data recorded. But at same moment it seemed necessary to interconnect more data of different fields for efficient planning. The goal for the planners was that the employees do not to lose track in a world that grew in complexity. The complaints about wrong or too much data were commonplace.

Referring to Western developments, the State Bank decided in 1974 to expand the information systems in scope through the implementation of remote data transmission. Like the Savings Bank of Stuttgart, they also ordered a report on data procession and transmission by their computer centre “VEB Datenverarbeitung der Finanzorgane” for defining the requirements. Based on that report, a nationwide transaction network had to be built transferring the data for an integrated information system in all banks called data collection system (DSS) – also in the Savings Banks. [57] There are two astonishing developments in respect to cross-border knowledge transfer between developers. First of all, the planners of the system had a very close look at the developments in the West. Horst Stevenson’s ideas are to be found word-by-word inside the reports, as well as the works of several other US-experts on banking automation. Besides rationalization, the argument of lagging behind the development in the capitalist countries should convince the state leaders of spending scarce resources in this project. As in the West, it was also the goal of the planners of the information system to record the data born digital. Through this the effort of data carrier transport and amount of the paper used should be reduced. Only in a second step, the recorded data should have been used for remote access in planning and control.

But, surprisingly enough, the data collection system was not developed only within the GDR. According to the before mentioned cooperation within the COMECON, the effort of developing the system was shared between socialist states

on a bilateral basis. On the Budapest trade fair in 1975, representatives of the State Bank of the GDR and the Hungarian company VIDEOTON, a big producer of information and communications technology met for negotiations. They reached an agreement about the construction of multiplex controllers for data transmission, seminars on remote data transmission and last but not least the programming of core components of the data collection system of the GDR. All in all, they agreed on software services. In the following years, experts from VIDEOTON travelled to the GDR on a regular base. Following the sources at the DP division of the State Bank, their employees were fully aware of the pitfalls of international cooperation in software projects. That's important because in general the COMECON is viewed as a failure. Head of division Uhlig stated that the quality of the *Systemunterlagen* depended upon coordination. He differentiated "The quality of the product depends on the skill of the programmer if he's working alone. If three programmers are working together, the quality of the product depends on project management". [58] This is a very early example of software metrics as the measurement of software production quality.

The international team of developers and planners faced three core challenges. First of all, it was necessary to establish an efficient project management. Therefore, the requirements had to be defined and the areas of competence had to be strictly assigned. The necessary effort of communication and coordination between the two countries was high, for example due to consultations. Program manager complained about the missing train and airplane connections from and to Budapest in the summer of 1976. Also the requirements tended to get out of hand. This caused Comrade Süßbier, head of the working group DSS, to change the requirements afterwards and he gave the order that "the requirements for the DSS are to be revised until the end of August. Everything that has nothing to do with the economic data collection system has to be thrown out. We need a clear line on that". [59] Secondly, delays in the production and in documentation were commonplace. For of this reason, general director Geißler of the DP-division sent an unequivocal letter to the representatives of VIDEOTON in Berlin in 1976 demanding a proper documentation. But this was not only a problem of international cooperation. Also within the GDR delays occurred quite often. For example, for test reasons the State Bank of the GDR ordered from the combine Zentronik modern teletypes T800 with a special coding. But the production of the teletypes came into delay so that the DP-division complained about this officially. But Dr Geißler only stated with brief words that "up to this day the general director of the people-owned combine Zentronik has not reacted to our letter despite several reminders". [60]

Thirdly, barriers of language had to be overcome. The programmers were united by one common language, the language of the machine. But despite this, the annotations as well as the knowledge of usage of the software were in very different languages. Software language depended on cultural styles, political contexts, historical paths, levels of development as well as on different believe systems. In the final code of the data collection system of the State Bank of the GDR, one can find five different languages. Most of the instructions were written in Assembler oriented

towards English as the *lingua franca* of the Digital Age. This wasn't changed by the politics of language of the GDR concerning imperialistic language. Besides English, many annotations are written in Hungarian, some others in German. Even French annotations are to be found. The French *Compagnie des Machines Bull* maintained good contacts into the Eastern Block. Adjacent to machine instructions like

```
DO or DATA
```

stood annotations like

```
*+SPACE
```

```
A HIGAS EGYSEG MEGJELOLESE // signs of defective units  
"ADRESSEN RETTEN" // rescue data
```

```
"TABLES DES ETIQUETTES DE CDS" // tables of CDS-labels  
[61]
```

While translators helped the programmers to communicate, on the machine level translators were not always available. Programs and computers from very different manufacturers of the ESER program had to talk to each other through protocols. This applied to the hardware level between computers and teletypes, for example in regards to the different currents for reception. [62] But also peripherals had to be connected to different types of machines. Robotron therefore produced an ESER-adapter to connect their own peripherals to the foreign ESER-computers. But in total, this took way more effort because of the necessary conversion processes. [63] Also on the software level, the problem was daunting despite the planned compatibility of ESER. So the head of sector DP, comrade Uhlig, was happy that at least there existed a way to emulate the software of VIDEOTON on their mainframe computers R10 – R40 in FORTRAN. [63] This was necessary to test the programs on the Robotron computers beforehand.

Despite the huge obstacles it seems likely that the international cooperation succeeded. Based on the newly built data network, the State Bank of the GDR in 1981 began to implement the DSS in the savings banks. Thereby, they circulated a preliminary draft of the working instructions for the preparation and training of the bank employees. In the following years, the instructions were adapted and extended several times. Even though the system had a rocky start, it went productive in 1983 and was used intensively quite fast. In the savings banks of the region Berlin, in 1986 alone, they processed 122.409 receipts with the DSS. [64] The savings banks in the GDR ordered ever more Teletypes to get connected to the system. The main obstacle was the bad transmission quality of a data network that was very error prone. And also the minds of the savings banks employees had to be opened for the new technology. In 1981, the DP division of the State Bank complained in a long letter about the problems of software implementation. After praising their own successes in data bank usability they stated that “it isn't always easy to get the new technology accepted. The change is about leaving behind traditional ways of work. Not always one can find the necessary openness to leave behind traditional behaviours”. [65] The implementation of software took its time.

In parallel with the nationwide implementation of the information system the international consultations about these systems continued within COMECON. [66] At a bilateral conference in 1981, the State Banks of the Soviet Union and the GDR exchanged their experiences about the use of information systems. The Soviet delegation was extremely interested in the developments inside the GDR. The very broad and unspecific terms the Soviet delegates used within these consultations seems to indicate that they had not reached a similar level yet. On the contrary, the GDR delegates had a deep insight into the technological developments of the USSR. They knew exactly which components they needed and named them openly as their main interest in a mutual exchange. [67] Eventually, ROBOTRON stuffed the computer center of the GOSBANK. [34] Information systems also were an important topic on the four working conferences on data processing of the State Banks of COMECON over the course of the 1980s. The first one took place in 1984, the second in Moscow in 1986, the third in Prague 1987 and the last one in Sofia in 1989. This went so far that after the third conference in Moscow the State Banks of Eastern Germany and the USSR agreed on a “proposal for the collaborative production and mutual exchange of software”. [68] This contained system software as well as applications. Even though both side came to the conclusion that the structure of their banking system differed widely in some areas, they saw overlapping use cases in human resources, documentation, back-office and control systems. [68] While both economic systems faced severe financial shortages at the end of the Cold War, the cooperative approach could be interpreted not only as rationalization of software production. Without doubt, it was a reaction to the widespread use of the Personal Computer at the work place that demanded a broader range of software. But it was also in the interest of the GDR to bind a USSR that withdrew its influence slowly under Gorbachev as it was confronted with an overstretch of power. [68]

3 Conclusion

Answering the initial question of this article, the bank was represented in its information systems step by step, covering finally every of its processes. The computerization of banking took place in four stages. After the introduction of Electronic Data Processing within singular fields of banking, Savings Banks intensified the efforts for data integration within these fields. The rise of information systems bridged the gap between the individual divisions and their different databases. Accessible via remote data transfer, information systems corresponded with the needs of management and employees for a better overview and the provision of the right data at the right moment of time. But their implementation faced severe obstacles. Even though software as the implementation of computers in banks never fulfilled the high expectations they aroused, systems like TELDAS in Stuttgart or the DSS of the State Bank of the GDR were in productive use since at least the 1970s. They combined different elements of the bank ranging from the computers, the employees up to the working instructions. Therefore, they built a Deleuzian

assemblage of those heterogeneous elements. During the course of the Cold War, the banks digitalized themselves as they finally in the 1980s depicted every one of their business processes in software. The Savings Banks now existed in the information system. That meant first of all a hardening of procedures. Processes and relations were defined by software that often ran for decades. Second, it made the bank visible and transparent to its management. Now it lay – within certain limits – open to them in real time what was going on inside the bank. Software was understood in a wider sense that included computer in action and their whole environment. In short: It was the code of banking that mattered.

As the whole project about the computerization of German Savings Banks is indicating, the plans of socialist countries often were even more ambitious and far reaching than in the West. That often meant that the GDR officials took the digitalization more serious for of different reasons. Apart from the demand for rationalization out of a severe resource shortage, it was mainly the desire for an efficient way of steering the economy under the imperative of a planned economy that drove the computerization in the banking sector. Or in a nutshell: The utopian character of socialism also showed itself in the digital (r)evolution. Obstacles were the huge delays in the production of information technology, the international cooperation and the poor quality of the data network provided by the Deutsche Post of the GDR. The information system of the State Bank finally was realized in 1981, the cybernetic plans for monitoring the economy were not. On the level of software development, the responsible leaders of the GDR took the COMECON serious. They placed their hopes on a socialist approach of cooperation within which the experiences of the computerization were shared. You can follow this up to the very code of the information system of the State Bank of the GDR that was written in five different languages. But like in West Germany the project suffered from severe shortages of software and programmers over the whole period of time. This caused again and again struggles of power and finally led to the strict standardization of the education and work of software engineers.

My final thesis is that socialist software in general transformed from a service to a product later than in the capitalist West. This was also influenced by the strong cooperation on the international level within the COMECON. But also the Savings Banks in West Germany took a rather open approach to software out of their understanding as oriented towards the common good. While in the beginning of the computerization the banks expected the huge hardware producers to deliver the software with their products or were self-produced, the software industry became ever more diversified since the late 1960s. A wider term of software not limited to the actual program but to the whole system of the computer in action helps to understand this change.

References

1. Nelson, T.: The Crafting of Media. In: Burnham, J., Robert, J.: Software. Information Technology: Its Meaning for Art. An Exhibition at the Jewish Museum, 17. New York, NY: The Jewish Museum (1970). P. 17.
2. Ensmenger, N.: The computer boys take over: computers, programmers, and the politics of technical expertise. (=History of computing) The MIT Press, Cambridge, Mass. (2010). P. 5.
3. For an entry point to this debate cf. Haigh, T., Priestley, M., Rope, C.: ENIAC in Action. Making and Remaking the Modern Computer. (=History of Computing) The MIT Press, Cambridge, Mass. (2016).
4. Cf. Kittler, F.: Grammophon, Film, Typewriter. Brinkmann & Bose, Berlin (1986). P. 7.
5. Cf. Mahoney, M.: What Makes the History of Software Hard. IEEE Annals of the History of Computing 30, no. 3, 8–18 (2008).
6. Barad, K.: Agential Realism: How Material-Discursive Practices Matter. In: *Signs* 28, no. 3, 803–31 (2003).
7. Cf. Schuhmann, A., Danyel, J.: Wege in Die Digitale Moderne. Computerisierung Als Gesellschaftlicher Wandel. In: Bösch, Frank (ed.): Geteilte Geschichte. Ost- Und Westdeutschland 1970–2000, p. 283–319. Vandenhoeck & Ruprecht, Göttingen (2015). P. 286-290.
8. In the international context cf. Kling, R., Iacono, S.: The Mobilization of Support for Computerization: The Role of Computerization Movements. *Social Problems* 35, no. 3, 226–43 (1988). P. 227.
9. Cf. Schmitt, M., Erdogan, J., Kasper, T., Funke, J.: Digitalgeschichte Deutschlands. Ein Forschungsbericht. *Technikgeschichte* 83, no. 1, 33–70 (2016). P. 33-34.
10. Haigh, T.: Software in the 1960s as Concept, Service, and Product. IEEE Annals of the History of Computing 24, no. 1, 5–13 (2002). P. 5-6.
11. Deleuze, G., Parnet, C.: *Dialogues II*. Columbia University Press, New York (2002). P. 69.
12. Cf. Leimbach, T.: Die Softwarebranche in Deutschland: Entwicklung eines Innovationssystems zwischen Forschung, Markt, Anwendung und Politik 1950 bis heute. Fraunhofer Verl., Stuttgart (2011). P. 122.
13. Gugerli, D.: Der Programmierer. In: Mangold, H., Frei, A. (eds.): *Das Personal der Postmoderne. Inventur einer Epoche*, p. 17–32. Transcript, Bielefeld (2015).
14. Wolle, S.: *Die heile Welt der Diktatur. Alltag und Herrschaft in der DDR 1971 – 1989*. 4. ed., Links Verlag, Berlin (2013). P. 86.
15. Donig, S.: Vorbild und Klassenfeind. Die USA und die DDR-Informatik in den 1960er Jahren. *Osteuropa* 59, no. 10, 89–100 (2009). P. 93.
16. Entry “Software”. In: Schoppan, W. (ed.): *Lexikon der Wirtschaft: Rechentechnik, Datenverarbeitung*, p. 364. Verl. Die Wirtschaft, Berlin (1981).
17. Campbell-Kelly, M.: *From Airline Reservations to Sonic the Hedgehog. A History of the Software Industry*. (=History of Computing) MIT Press, Cambridge, Mass. (2004). Here P. 2.
18. Cf. Mirowski, P.: *Machine Dreams. Economics becomes a Cyborg Science*. Cambridge Univ. Press, Cambridge, UK (2003).
19. Ronald R.: Address at Moscow State University, May 31, 1988 <http://millercenter.org/president/reagan/speeches/speech-3416>.

20. Cf. Chun, W.: Programmed visions. Software and memory. (=Software studies) MIT Press, Cambridge, Mass. (2013). p. 4-6.
21. Mahoney, M.: The History of Computing in the History of Technology. *Annals of the History of Computing* 10, no. 2, 113–125 (1988). P. 122.
22. Cf. Deleuze, G., Guattari, F.: *A Thousand Plateaus. Capitalism and Schizophrenia*. University of Minnesota Press, Minneapolis (1987). P. 3-28.
23. Rosenberg, S.: *Dreaming in Code. Two Dozen Programmers, Three Years, 4,732 Bugs, and One Quest for Transcendent Software*. New York: Crown Publishers (2007).
24. WABW B102/2300. IBM Deutschland: Elektronische Datenverarbeitung Bei Der Städtischen Sparkasse Stuttgart, Städtische Girokasse Stuttgart. Ein Untersuchungsbericht Der IBM Deutschland (1960).
25. Cf. Martin, I.: Structuring Information Work. *Ferranti and Martins Bank, 1952-1968. Information & Culture: A Journal of History* 47, no. 3, 312–39 (2012). P. 325.
26. WABW B102/321. Städtische Girokasse Stuttgart: Austausch der elektronischen Datenverarbeitungsanlagen vom Typ IBM 1401 gegen die Type IBM 1460 (1963). P. 6.
27. Ganzhorn, Karl (ed.): *IBM Informationstechnik für Banken und Sparkassen im 20. Jahrhundert. Forschung und Entwicklung in der IBM Deutschland* 6. Röh, Sindelfingen (2006). P. 51.
28. WABW B102/321. Städtische Girokasse Stuttgart: Beschaffung einer neuen Datenverarbeitungsanlage (1967). This and all following quotations translated by the author.
29. Interview with Horst Zuse, son of Konrad Zuse conducted on 05/17/2016.
30. For the Nixdorf 820 see Berg, Christian. *Heinz Nixdorf. Eine Biographie*. Ferdinand Schöningh, Paderborn (2016). P. 94-99.
31. Ganzhorn, Karl. *The IBM Laboratories in Boeblingen. Foundation and Build-Up*. Röh, Sindelfingen (2000).
32. Interview with Winfried Ferger, conducted on 09/04/2015 at IBM Deutschland.
33. WABW B102/1451.
34. Interview with Hans-Jürgen Lodahl, conducted on 06/13/2016.
35. Ulbricht, Walter: Rede Des Genossen Ulbricht Vor Dem Bezirkspateiaktiv Halle. In: *Neues Deutschland* (November 14, 1966). P. 3.
36. Cf. Gräßler, R.: Die Entwicklung von Sachgebietsorientierten Programmiersystemen ‘SOPS’ Des VEB Kombines Robotron. In: Naumann, F., Schade, G. (eds.): *Informatik in Der DDR - Eine Bilanz*, 288–300. GI, Bonn (2006).
37. Cf. Geipel, G.: Politics and Computers in the Honecker Era. In: Mackris, Kristie, Hoffmann, Dieter (eds.): *Science under Socialism*, 230–246. Harvard Univ. Press, Cambridge, Mass., (1999).
38. On COMECON and its failure cf. Ahrens, R.: *Gegenseitige Wirtschaftshilfe? Die DDR im RGW: Strukturen und handelspolitische Strategien 1963-1976*. Böhlau, Köln, 2000.
39. Fritsche, D.: Mit Prototyprekonstruktion zum Welthöchststand? PC-Software in den letzten Jahren der DDR. In: *Dresdener Beiträge zur Geschichte der Technikwissenschaften* 30, 105–123 (2005).
40. Stevenson, H.: *Informationssysteme für Kreditinstitute. (=IS Informations-Systeme)* de Gruyter, Berlin, New York (1973). P. 13.
41. Doering-Manteuffel, A., Lutz R.: *Nach Dem Boom. Perspektiven Auf Die Zeitgeschichte Seit 1970*. Vandenhoeck & Ruprecht, Göttingen (2008).
42. Cf. Cortada, J.: *The Information Flood*. Oxford University Press, Oxford, New York (2012)

43. On the fact that data overload isn't a recent phenomenon, see Weinberger, D.: Too Big to Know. Basic Books, New York (2012).
44. Haigh, T.: Inventing Information Systems. The Systems Men and the Computer, 1950-1968. In: Business History Review 75, 15–61 (2001).
45. Dahms, H., Haberlandt, K.: Erfahrungen und Grundsätze beim Aufbau eines automatisierten MIS. IO: Zeitschrift für Betriebswissenschaft, Management, Produktionstechnik u. Organisation 39, no. 11, 449–459 (1970). P. 456.
46. Cf. Gießmann, S., Marcus B.: Was Ist Datenkritik? Zur Einführung. In: Mediale Kontrolle Unter Beobachtung 3, no. 1 (2014).
47. Cf. Holtzmann, I., Stieffenhofer, H.: SODIS. Sofortdispositions- und Informationssystem der Sparkasse der Stadt Saarbrücken. IBM-Nachrichten 18, no. 187 (1967), p. 44–50.
48. Der Bankmann und sein Computer. Zeitschrift für das gesamte Kreditwesen 20, no. 21: 977–978 (1967). P. 978.
49. Ein EDV-System feiert Geburtstag. 10 Jahre Online-Betrieb bei der Stadtparkasse Saarbrücken. Computerwoche (11.11.1977), <http://www.computerwoche.de/a/10-jahre-online-betrieb-bei-der-stadtparkasse-saarbruecken,1200155>
50. Cf. Internationales Institut Der Sparkassen: 3. Internationale Automationstagung, Vedbaek, Mai 1967. Amsterdam (1967).
51. Internationales Institut der Sparkassen: Stand Der Automation Und Der Elektronischen Datenverarbeitung Anfang 1973. Genf (1973). P. 41.
52. Cf. Belvederesi-Kochs, R.: Von der "moralischen Anstalt" zum vertriebsorientierten Finanzdienstleister. Der unternehmenskulturelle Wandel des Deutschen Sparkassen- und Giroverbands im Spiegel seiner Marketingstrategie. Zeitschrift für Unternehmensgeschichte 53, No. 2: 192–215 (2008).
53. Cf. Agar, J.: The government machine: a revolutionary history of the computer. MIT Press, Cambridge, Mass. (2003).
54. WABW B102/638. Landesgirokasse Stuttgart: 50 Jahre Datenverarbeitung bei der Landesgirokasse. In: Landesgirokasse Stuttgart: Bericht über das Geschäftsjahr 1980. P. 48–50 (1980).
55. Internationales Institut der Sparkassen: Automation Trends 1979. Internationale Untersuchung Der Trends Und Neue Entwicklung Bei Sparkassen. Genf (1979).
56. Anwendergemeinschaft EDV-Projekt "Kontokorrent, Verrechnungsverkehr und Bankbuchhaltung sowie Automatisierung des Zahlungsverkehrs": Anschreiben 3. Nachtrag zum Handbuch EDV. In: Staatsbank der DDR: Handbuch der Sparkassen 5-53: EDV-Projekt Kontokorrent, Verrechnungsverkehr und Bankbuchhaltung. Berlin (1971).
57. BArch DN10/633. Staatsbank der DDR, Abteilung Datenverarbeitung: 1. Entwurf einer Studie zum Problem der Datenfernverarbeitung und Datenfernverarbeitung im Kreditsystem der DDR (1974).
58. BArch DN10/633. Staatsbank der DDR, Abteilung EDV: Einzelnotizen auf Grund von Beratungen und Gesprächen mit Vertretern der VIDEOTON-Fabriken und der VIDEOTON AG anlässlich der Budapester Messe 1975, (1975). P. 4.
59. BArch DN10/633. Staatsbank der DDR, Arbeitsgruppe DSS: Stand der Arbeiten am DSS: Probleme (1976).
60. BArch DN10/633. Staatsbank der DDR, Abteilung EDV: Problemzusammenstellung für das Ministerium für Elektrotechnik/Elektronik (1975). P. 3.
61. BArch DN10/633. Code.
62. BArch DN10/633. Staatsbank der DDR, Abteilung EDV: Brief des Generaldirektors Geißler an VIDEOTON (1967).

63. BArch DN10/633. Staatsbank der DDR, Abteilung EDV: Einzelnotizen auf Grund von Beratungen und Gesprächen mit Vertretern der VIDEOTON-Fabriken und der VIDEOTON AG anlässlich der Budapester Messe (1975). P. 4.
64. BArch DN1/20907. Ministerium der Finanzen/Abteilung Datenverarbeitung: EDV-Anforderungen und -Aufgabenstellungen. Einführung des Datensammelsystems in den Banken und Sparkassen, 1980-1988 (1980).
65. BArch DN10/1909. Staatsbank der DDR, Abteilungsbereich 602: Bericht anlässlich des Erfahrungsaustausches von EDV-Spezialisten der Staatsbanken der RGW-Länder am 4./5.11.1981 in Budapest (1981). P. 9 .
66. BArch DN1/17439. Ministerium der Finanzen, Kaminsky: Brief an den Oberbürgermeister der Hauptstadt der DDR (1970). P. 2.
67. BArch DN10/1909. Staatsbank der DDR: Internationale Arbeit zu EDV mit Schwerpunkt Informationssystem (1981).
68. BArch DN10/1909. Staatsbank der DDR, Abteilung 63: Maßnahmenplan in Auswertung der zweiten EDV-Konferenz vom 12.1.1987 (1987).