Tensions of Europe/Inventing Europe

Working paper nr. 2008_03

Title: Software for Europe:
Constructing Europe through Software

Author: Gerard Alberts et al.

Date: October 2008

Keywords:

Abstract:

Software — the instructions allowing computers to be used — has played a significant and under-explored role in the shaping of postwar Europe. We address this role through examining the tensions between two contrasting modes of computer technology appropriation: the direct importation of applications software, as promoted by the practices of corporations such as IBM; and the development of software through university-industry co-entrepreneurship, demonstrated in the case of the programming language ALGOL.

Writing a contextual history of software allows us to address historical themes concerning Europe and Europeanness in the second half of the twentieth century. In the initial era of postwar reconstruction, with declinist rhetoric setting in across much of Europe, building a computing machine could be seen as a source of specifically national pride; a decade later, the shift from hardware to software initiatives appeared to present a very different, universalist and internationalist character. What informed this change? Underlying software standards, meanwhile, promoted a largely hidden mode of integration: what was its role in the move towards European unification and the Cold War?

As a collaborative research project, building on growing cooperation in European history of computing, Software for Europe aims to answer these questions.

Software in the shaping of Europe

Software underlies information management and communication worldwide. Its quiet pervasiveness provides an ideal case for integrating a study in the history of technology into histories of the broadest scope. Electronic data-processing increasingly signals connection to the events that matter from a political, cultural or economic perspective; its absence signals exclusion. Beyond simply facilitating the functioning of everyday life, software and its producers and users define, in many ways, the categories through which the world is understood.

A single example may serve to illustrate the contribution of software to standardization, and thus to integration in Europe. Enterprise Resource Planning software (ERP), widely adopted by European corporations, is a category of huge software packages controlling and managing the entire organization of a firm: its finances, its personnel, its material flows and stocks. Such packages embody a rationalized business model, their adoption enforcing standardization in business cultures. The concept of ERP software as a market niche was in part created in the 1990s by European software houses, like SAP in Germany and BAAN in the Netherlands. How did standardization processes like this take place? Were such processes different in Western, Central, Eastern and Nordic Europe? Has software in its pervasiveness played integrating or dis-integrating roles, either hidden or overt?

Objectives

"Software for Europe" is a collaborative research project on the role of information technology (IT), and in particular of computer software, in the history of Europe. We use software as a lens to focus on the relations between information technology use and the shaping of European policies and infrastructures, examine whether there have been specifically 'European' styles or modes of working in software development, and to what extent software practices have contributed to reinventions of Europe.

The present historiography of computing allows us to start from sound foundations in the early machine era to develop a clear focus on the rise of software in the 1960s, addressing the software industry in the 1970s, and go on to address factors responsible for the state of our present-day knowledge society.

Our main theme within "Inventing Europe" is constructing European ways of knowing. Formal initiatives, such as the ALGOL effort (see below), and informal, hitherto unaddressed connections between computer scientists across international boundaries, contributed to the development of techniques and standards distinct from those emerging elsewhere in the world. We also address the themes of building Europe through infrastructures (via the creation of these formal and informal networks and their consequences in the growth of information-processing cultures), consuming Europe (studying the adoption of technologies such as programming languages and applications software products by a variety of customers), and Europe in the global world (through the interactions of the Marshall Plan and Cold War with the development of software professions and industries within the nations of Europe, and the UNESCO-sponsored International Federation for Information Processing as a space for the negotiation of European identity.)

The Software for Europe project uses software as a lens to focus on the relationship between computer use, software development and the shaping of Europe. We identify two major modes of appropriation of information technology, which will be traced and contrasted across the second half of the twentieth century. The first, artefact appropriation, involves the adoption and mediation of pre-existing technical systems and the visions and practices of use associated with them. The second mode, concept appropriation, implies the adoption of shared goals at the conceptual or theoretical level, where the development of hardware and software may vary locally. These two modes are not understood as informing disjoint traditions. They meet, intertwine, clash, part and meet again: they define a tension. Software for Europe addresses the multifarious consequences of this tension, and its role in the definition and establishment of 'European' and 'non-European' character in information-technology development.

Artefact appropriation: IBM and Europe

The first mode of appropriation has been most common in data-processing in the fields of banking, insurance and civil service. European national markets were typically dominated from the 1950s by US corporations, most notably International Business Machines (IBM), which operated on a vastly greater scale than its competitors and was the most strongly perceived as "American" in character. Cliché has it that IBM's clients were encouraged to follow a monolithic corporate culture, including the scripts of its machinery: our aim here is to question this, pointing both to the agency of national users and to the multiplicity of meanings resulting from IBM's policy of local assimilation. In some countries, including Finland, IBM stood for international progressiveness, acting as an entry-point to Europe as it took its prospective clients to Stockholm or Paris; in the Netherlands, Belgium or France, by contrast, IBM rather symbolized American culture, even if clients travelled to Paris or to Stuttgart to see the latest models. The IBM users' organization, SHARE, was renowned for its influence on company policy and on the nature of IBM software as it developed. 1966 saw the foundation of an affiliate group, SHARE European Association (SEAS.) In European eyes, the very same adoption of technology might appear as an entry point to modern life or as conservative business imposing itself, as progressive western or decadent American culture. Perceptions of the relations between modernity, modern technology and American culture were ambiguous (cf Misa, Brey and Feenberg 2003). Was there, beyond the symbolic and commercial role of IBM and its competitors, a hidden integrating, and perhaps at the same time dis-integrating influence in the technology and policy of these actors?

Concept appropriation: ALGOL and the European space for software

The explicit appropriation of shared ideas about computers, as distinct from the artefacts, is visible from the late 1940s in a variety of local initiatives grounded in established collaborative cultures of measuring and computing; here, the need for software played a key role. Whereas, in the USA, a commercial software sector had identifiably emerged by 1958 (Campbell-Kelly 2003), Europe presents an under-explored case in which no such sector existed. Typically, the computers manufactured in various European countries would be delivered without software; the task of writing code, compilers and operating systems, was taken on by academic teams outside the pre-existing commercial sphere. This pattern was seen in Amsterdam, Grenoble, Mainz, Munich, Vienna and Copenhagen. If this entrepreneurial spirit defied the academic convention of staying out of the muddle of private interest, the computer specialists may have acted as a counterculture; or perhaps the academic habitus was not as unambiguous as the European self-image would have it. In 1959, UNESCO capitalized on the established integrative tendency with the formation of what became the International Federation for Information Processing (IFIP), an umbrella for national organizations and a forum for collaborative activity. Though its remit was global, IFIP is remembered for a number of initiatives with strong European traits. Most notable of these is ALGOL (for ALGOrithmic Language), an early example of a high-level programming language, used to communicate with machines in terms convenient and accessible to human operators. The "purity" of the mathematically-refined ALGOL is widely contrasted, in received opinion, with the less elegant but more widely-applied language FORTRAN, a product of IBM's US-focused corporate culture.

How was this ambiguity negotiated — could the cultures promoted by UNESCO, IFIP and ALGOL be both European and global? Did national funding agencies promote the construction of particular images? Was the "ALGOL effort" dominant and centralizing within Europe? Software for Europe proposes as a working hypothesis that, beyond the effort to define a new language, the culture of software co-entrepreneurship across borders represented by ALGOL helped to create a specifically European space for software.

Studying Europe

Software for Europe aims to trace the history of software, not in isolation, but in order better to understanding the shaping of European infrastructure. This will be addressed at a number of levels, representing institutional, material and discursive considerations.

Europe through formal and informal organizations

Ideas informing the direction of software development circulated through user groups such as SEAS, professional bodies like IFIP, and professional conferences sponsored by these and other organizations.

What is commonly regarded as the single most important meeting in the history of software took place in Europe: in 1968, leading figures from industry and academia convened in the Bavarian ski resort of Garmisch Partenkirchen for a NATO-sponsored under the title “Software Engineering.” Participants reported, with remarkable mutual candour, a widespread inability to manage large software projects effectively with the resources available. This “software crisis” was co-constructed with a variety of proposed remedies emphasizing the need for standards, greater formalism, and the levels of systematic training associated with a mature, professionalised engineering discipline. Garmisch Partenkirchen rapidly attained the status of a watershed in software identity, and the “Software Engineering” position became prominent in academic curricula, to some extent displacing more mathematically-based practices. Software for Europe will examine the extent to which this broad institutional change influenced the emerging software industry in European countries: Garmisch was on several levels a European event, and played a key role in the creation of a European space for software.

European patterns of innovation

The ALGOL initiative, originated around 1955, was rather different in its origins and expectations: it may be viewed as an academically-grounded attempt to impose on industry not merely an operating standard, but a whole research agenda imposing norms at a more general and overt level. The initiative sought to change (in its proponents’ view, to correct) US-originated, and particularly IBM-based cultures of computer usage, and in this respect fell far short of its target; the IBM user organization SHARE, in fact, provided one of the strongest mobilizing points for forces opposing ALGOL. The distinctive co-entrepreneurship patterns embodied in the ALGOL effort, then, exhibit a mode of negotiation between academia and industry which was commonly perceived as non-American and, typically, as more specifically European. Notwithstanding opposition, ALGOL was adopted worldwide as an official publication standard for computer programs, while computer science students in European universities were introduced to programming through ALGOL, and later its similarly “elegant” descendant PASCAL. We address the tensions between European and global identity in the ALGOL effort, and the question of whether it generated a path-dependency which distinguished the direction European computer science from that developing elsewhere.

European themes in history

NATO’s sponsorship of the Garmisch conference, an entirely unexceptional development, serves as a reminder of the Cold War context in which the emergence of professionalized computer science and software engineering took place. The “hidden integration” which is such a remarkable feature of the software case, however, forces us to go beyond simple notions of Eastern and Western blocs in isolation, or of a US hegemony straightforwardly dictating Western cultural norms. The increasingly broad adoption of IBM standards and machinery — overtly in Western Europe, and covertly in the East, where local initiatives relied on what would now be regarded as unauthorized “cloning” — suggests a different perspective, as does the and the alternative, conceptually-focused pattern of cooperation enabled by the ALGOL effort. A related theme is the status of Europe in the eyes of non-Europeans: we will investigate IBM’s perceptions of European markets and cultures, and their practical consequences. Again, the concept of tension is instrumental: the varying patterns of collaboration and antagonism between the US corporation and European actors inform several of the partners’ projects.

State of the history of computing

The Software for Europe project is positioned at the cutting edge of the history of computing. Reflecting wider developments in the history of technology, a traditional focus on production-focused, hardware-based narratives is now giving way to nuanced contextual accounts highlighting the crucial importance of software, its mediation and use. Recent collaborations in the field (Hashagen et al 2002) have prepared the ground for a conscious historiographic shift towards this position, codified in the work of Martin Campbell-Kelly (2003) and Thomas Haigh (2001. 2002) and permitting for the first time a productive historical location and identification of the term “software” (Alberts, Bogaard, Campbell-Kelly and Veraart, 2004.) Software is not only instruction code, but a category of systems understood since the late 1950s as economic goods as well as embodiments of technical and policy positions, and signals with striking clarity the divergence between US and European histories.

Whereas, in the USA, a service and consultancy sector (currently the subject of extensive research by Yost and others) grew to produce such software, European nations typically favoured less formal and less evidently commercialized patterns of academic-industrial co-entrepreneurship. Where US technologies were adopted, as in the IBM case, there was extensive local variation in presentation and understandings. “Software for Europe” will provide a necessary corrective to a historiography so far dominated by US-centred accounts, at the same time revealing the interplay of diverse agencies and the multiplicity of possibilities in the development of information culture.

The project offers a coherent set of studies framed by the tension between the two modes of appropriation. Building on established cooperations between several of the partners in recent years, we ensure the coherence of the collaboration and stimulate the circulation of ideas through a pre-planned series of winter workshops and summer schools, to be hosted by each of the participants in turn.

Method

Software for Europe is a historical project with strong interdisciplinary connections. Its members are informed variously by the disciplinary perspectives of cultural history, business history, economic history, history of science and technology, science and technology studies, and technology policy; it is our intention that the project’s work will be recognised in all these fields. Much of the work will proceed from analysis of written sources, published and archival, and to some extent on the examination of software itself (a practice not yet widely developed, to which the project is expected to contribute some methodological innovation.) Evidence will also be drawn from interviews with policymakers and representative users from the relevant industrial, academic, political and administrative constituencies.

Using IBM

Bringing together historical expertise on computer usage, on IBM Europe and its user organisations would not even have been concieved of without Software for Europe. Given the expertise of Paju, Leimbach, Van den Bogaard, Mounier-Kuhn, Mols, Sumner on using computers in business context we may indeed expect to cultivate this unexplored field and simultanuously tell the story of shaping Europe, through hidden integration and desintegration, and contribute in an important way to the state of history of computing. It goes without saying that cooperation will be sought with similar research at the University of Pennsylvania (Ensmenger, Schlombs) on user groups in the US. As an extra, Durnova and Paju look in a collaborative effort into the role of IBM in countries in direct vicinity of the Soviet Union.

ALGOL effort

Durnova, Mounier-Kuhn, Mols, Sumner, Seising, Alberts and Nofre will combine their intimate knowledge of numerical analysis and the mathematical side of computing to investigate the ALGOL effort in general –taking the historiography from the numerous retrospective writings of the pioneers to an historical account– and in their fresh research rigorously focus on that curious European pattern of university-industry co-entrepreneurship (VUMS in Prague; IMAG in Grenoble; MBLE in Brussels; construction Autocode and programming tools in Cambridge and Manchester; Compiler construction at the Mathematical Centre in Amsterdam). As to the investigation of German and Danish varieties of the co-entrepreneurship (ALCOR and Regnecentralen), Alberts has made himself familiar with the sources and secondary literature far enough to allow us to include these in the synopsis.

Both the intellectual openness to look at the two phenomena, and the actual overlap of the groups of prime experts, will guarantee that the two modes of appropriation indicated here will not be studied in isolation but in their tense interplay.

Emergence of software

Implementing data processing machinery, even before computers, involved considerable investments in consultancy and service. The content of this phenomenon has been aptly described by Thomas Haigh, showing the continuity in the work of the "systems men" in putting IBM machinery to use. In historical surveys of the rise of the service and software industry Martin Campbell-Kelly, James Cortada and Jeff Yost have offered views of the economic appearance of the same phenomenon. Associated projects by Haigh and Yost, building upon this work, make sure the economic and cultural role of this industry is well covered in the CRP Software for Europe. Alberts, Van den Bogaard and Leimbach at different occasions in discussion with Campbell-Kelly and Haigh have stressed the issue of emergence of software in the late 1950s.

Analysing notions like "appropriation" and historicizing concepts like "software" will be an integral part of the collaborative learning process of Software for Europe. Here is a proper basis to allow our team to historisise the notion of software as we proceed, by consequence research the history of software for Europe beyond naiveté.

Individual projects

1. Electrologica's software?
   Gerard Alberts (Project Leader), University of Amsterdam, G.Alberts@uva.nl
   Adrienne van den Bogaard (Postdoc), TU Delft
   David Nofre Mateo (PhD candidate), Amsterdam

2. Czech(oslovak) participation in the ALGOL effort
   Helena Durnova, Brno University of Technology, durnova@feec.vutbr.cz

3. Using IBM in Europe to recapture the lead?
   Hannu Salmi, University of Turku, hansalmi@utu.fi
   Petri Paju (Postdoc), University of Turku

Associated projects

4. Competing Modernities: 'Machines Américaines', Fortran & Algol in Belgium, 1940-1968
   Sandra Mols, Namur, sandramols@yahoo.co.uk

5. Software Development Configurations: The 'IMAG' Lab at Grenoble and the Computer Industry
   Pierre Mounier-Kuhn,Sorbonne, Paris, mounier@msh-paris.fr

6. Software in Britain — computer appropriation, automatic coding
   and the problematics of the 'British Problem,' 1948-1970
   James Sumner, University of Manchester, james.sumner@manchester.ac.uk
n.n. (Postdoc), Manchester

7. Software Tensions in Non-Anglophone European Contexts
   Aristotle Tympas, National and Kapodistrian University of Athens, tympas@phs.uoa.gr
   Theodore Lekkas and Dimitris Ziakkas (PhD candidates), Athens

8. ICT and Business. The Rise and Development of Software-based Industries in Europe
   Paul Erker, Deutsches Museum, Munich, p.erker@deutsches-museum.de
   Timo Leimbach (Postdoc), Munich

9. Genesis and Development of "Soft Computing / Computational Intelligence"
   Rudolf Seising, Munich, rudolf.seising@meduniwien.ac.at

10. Inside the Box: A History of the Software Package
   Thomas Haigh, University of Wisconsin, Milwaukee thaigh@computer.org

11. History of the Computer Services Industry
   Jeff Yost, Charles Babbage Institute, University of Minnesota, yostx003@tc.umn.edu

Individual project 1

Electrologica's software?
Co-entrepreneurship and the emergence of a Dutch software industry

Gerard Alberts, Amsterdam
Adrienne van den Bogaard, Delft
David Nofre Mateo (PhD candidate), Amsterdam

The phenomenon that Dutch computer manufacturer Electrologica by 1960 did not have a software department is brought to bear in two directions.

a. Academic researchers jumping in and developing a university-industry co-entrepreneurship for the construction of compilers and operating systems did not present an exception but reflected a European pattern of early software production.

b. Electrologica in due time did develop a software department. In fact, 10 years later the software sector was an emerging economic sector in The Netherlands, having strong roots in IBM software practices.

c. In service of the entire Software for Europe CRP some budget is asked to facilitate the steering and management of SOFT - EU

Electrologica was a successful Dutch computer manufacturer, from 1958 onwards delivering its X1 machines as bare hardware. The firm did not have a software department. Instead university groups (Delft, Leiden, Utrecht and the Amsterdam Mathematisch Centrum) took the responsibility and constructed compilers (ALGOL in particular), operating systems and other programs. Curious as this situation may seem, it was the general pattern for European computer manufacturers. Indeed industry and academic computing centres were happy, in defiance of the existing compact of non-interference between science and society (Smith 1994), to lean on each other, as imperfect institutions longing for their complement. Their co-entrepreneurship during the 1960s created a European space for software, creating international cooperation and serving as a bedding for the emerging software industry. We will study the Dutch side of this story in European perspective and study the rise of a software industry in The Netherlands.

a. ALGOL compilers: university-industry co-entrepreneurship

David Nofre Mateo (PhD candidate) and Gerard Alberts (supervision)

Edsger Dijkstra and Jaap Zonneveld, at the Mathematisch Centrum of Amsterdam, gained instant fame within the international ALGOL community when in August 1960 they were the first to deliver a compiler for the full language of ALGOL. A compiler is a translator: a program automatically translating from a "higher" programming language to machine code (Kruseman 2003). Given the controversies and the voting on what to include in the language, just a few months before on a meeting in Paris, this was quite a feat. Dijkstra went on to become the expert in the construction of compilers and operating systems, on programming at the point in history where programming was turning into software (Dijkstra 1962). The British volume on *ALGOL 60 implementation* has his Foreword (Randell, Russell, 1964); when the compiler group of the Copenhagen Regnecentralen sought proof for the quality of its ALGOL compiler, it invoked his decree calling it a masterpiece (Bech 1975, p 68). These influences immediately illustrate that ALGOL, much more than a programming language stood symbol for an effort, an intellectual agenda, or even, in the words of one of its protagonists a "conspiracy" (Bauer 2004). The ALGOL effort was truly international, with a strong European emphasis.

One line of research will be to straightforwardly trace the history of the ALGOL effort, on which primary sources and retrospective literature is abound (Bemer 1969, Naur 1978, Perlis 1978), and treat it as a historical subject (Mahoney 1988, 1990, 2002). The aim is to describe the ALGOL side of the European space for software.

The second line of research is to exploit the fact that academic researchers were constructing compilers for a commercially available computer, Electrologica X1, that came without programs. Such cooperation, that synchronically occurred in Mainz and Munich (ALCOR group), Copenhagen (Regnecentralen's Compiler group), Grenoble (IMAG), looks like a typical European pattern of co-entrepreneurship. The Dutch version will be studied in detail: who cooperated, what were the material and immaterial rewards, what other interests were served, what were the relations to commercial programming efforts, what about the "purity" of science? The compact between science and society, reigning in The Netherlands just as in the US (Smith 1994, Haigh 1995), was enforced by national science policies and would normally withhold academics from such cooperation. Perhaps the ideology was not as strong in practice as in the official documents; or were there ways about?

The main claim to be researched in detail is that this pattern of co-entrepreneurship has served as the European, at least the Dutch, entry from programming into software. The content of the cooperation was the construction of compilers and operating systems: these products were historically the items to be called software. If we are right in interpreting the academic-industrial co-entrepreneurship as the locus of constructing what in the US was predominantly an industrial enterprise, then the study of these details will show the birch of a European space for software. The result will give us European ways of knowing, infrastructures (formal and informal organisations) and consumption junctions (patterns of co-construction) in one.

b. Sites of tension between IBM software culture and European software practices: Electrologica, Philips, and IBM-Netherlands

Adrienne van den Bogaard

Electrologica did develop its own software department. In fact Philips Computer Industry, the Philips branch founded in 1963, which took over Electrologica in 1968, had from the start recognized the importance of software and hired Roy Nutt, from US based Computer Sciences Corporation, to advise on software development. The Philips Electrologica connection is taken here as an entry point to study the software development practices in the emerging Dutch software industry. Alternatively these practices may be seen as emerging either by copying IBM software culture or resisting it.

Clients of Electrologica were universities and industries all over Europe (example: Hoechst). Software for the university customers was constructed by academics, typically at the forefront of the ALGOL effort, interested in large scale scientific computation and the supporting software to facilitate such application. Industrial clients, by contrast, had established data processing practices (with punch card machines as their core technology) and were already accustomed to systems thinking. These had little interest in the ALGOL effort, as they depended on continuity in data-processing practices or laboratory practices. Electrologica's software department, once established and growing to a hundred programmers by 1963, would provide programs, compilers and applications in FORTRAN and other languages. Electrologica is the perfect case to study the tensions between American and European software construction.

The presence of IBM software culture goes back to the interbellum when Dutch systems men, like R. W. Starreveld, introduced systems thinking into Dutch corporate management. When in the late 1960s a Dutch software industry emerged, IBM-Netherlands was the biggest software house. This project studies the influence of IBM on Dutch software practices by 1. tracing IBM culture in newly founded software houses (e.g. Volmac) 2. tracing resistance to IBM culture in newly founded software houses (example BSO) 3. by investigating the role of the Dutch contribution to the European User Organization of SHARE (the SEAS).

Individual project 2

Czech(oslovak) participation in the ALGOL effort

Helena Durnová, Brno University of Technology

Software/programming efforts in Czechoslovakia seem to follow a pattern different (and in a sense more European) from the development of hardware. Initially a part of mathematics (hence also the name “mathematical informatics”), these efforts were not under such a strong control as the calculating machines. In spite of the officially declared “bourgeois” character of cybernetics, the intellectual endeavour connected with software (ALGOL being a prominent example) found its niche within the community of mathematicians, and could thus demonstrate the pervasiveness of European ways of constructing knowledge even under the Communist Party regime.

After the „Velvet Revolution“ of November 1989 leading Czechoslovak intellectuals would talk of a „return to Europe“, to which others reacted by the geographically justified statement that Czechoslovakia (since 1993, the Czech and the Slovak Republics) had never ceased being part of Europe. Geographical position, however, can hardly be seen as the only sign of Europeanness.

As the country lay directly on the border drawn by the Cold War, it had first-hand experience of the divide – a divide that needs to be taken into account when discussing the shaping of post-war Europe. While relatively tight with regard to material transfer, the border seemed much more permeable with regard to institutions and discourse. Research in the field of software and computing technology suggests different level of control over the actual computing machines on one hand and over the spread of programming practices on the other: while there were little objections to software and programming, computers imported from the West were not welcome by the establishment. The proposed project will aim at clarifying the difference in approach between the material (computer) and discursive (programming) levels.

Socio-economic-political background

After declining participation in the Marshall plan and especially after joining Comecon in January 1949, Czechoslovakia gradually lost the economic ties to the countries on the other side of the Iron Curtain (cf. Kaplan, 1995). This, however, does not mean that all intellectual ties were broken (cf. recollections by the Austrian computer pioneer Heinz Zemanek of his friendship with his Czech counterpart Antonín Svoboda --before Svoboda left the country for the USA in 1964).

The Prague Spring of 1968 and its suppression paradoxically signified a turning point in computer science as well. The start of efforts to build a “unified system of electronic computers” JSEP (Jednotný systém elektronických počítačů) in 1968 preceded the grey, “normalized”, life of the 1970s. These computers were called “EC”, which is the (Cyrillic alphabet) abbreviation of the Russian phrase for “jedinstvennaja sistema”, or “the unified system”. The joint effort of the Soviet Union, Poland, Hungary, Bulgaria, Eastern Germany and Czechoslovakia, with other Comecon countries joining later, worked until 1989. Although it seems to be common knowledge among programmers that the JSEP computers were unauthorised copies of IBM computers, in the official publications emphasis was put on the *compatibility* with IBM products. Similarly to Finland (cf. the individual project of Salmi and Paju), IBM stood for quality, albeit only outside the official declarations in the case of Czechoslovakia. Was computer usage a way to protest against one of the two world powers, the USA and the Soviet Union?

After the relaxed atmosphere of the 1960s, the 1970s in Czechoslovakia witnessed the brain drain following the 1968 invasion of Soviet, Hungarian, Polish, Bulgarian, and Eastern German troops. The losses were of three kinds: some people left the country, others left “just” their positions, leaving the remaining ones in an atmosphere of fear of losing theirs. The climate of fear to stand out permeated the country (cf. Vopěnka). It is often believed that science departments at universities and polytechnics in general were not under such strong control of the regime as e.g. faculties of arts. Computing technology and software seem to present a special case: while there are little objections to software and programming, computer hardware presents a major problem, especially when imported from the West.

The proposed project will aim at clarifying the difference in approach between the material (computer) and discursive (programming) levels.

History of computing technology in Czechoslovakia

Historiography of computing in Czechoslovakia has been largely confined to eyewitness reports (cf. Korvasová, Jiřina, Folta) with parsimonious attention to software.

Most of the hardware development is connected with the Research Institute for Mathematical Machines (in Czech referred to as VÚMS), whose beginnings can be traced to the circle of Antonín Svoboda (1907-1980) in the late 1940s. In 1950, the "Laboratory for Mathematical Machines" was founded as a part of the Central Institute for Mathematics of the Bohemian Academy of Sciences and Arts. Antonín Svoboda led this Laboratory working on the project of an automatic computer, which was crowned by success in 1957, when SAPO (SAmočinný POčítač, i.e. automatic computer) was put into operation. In 1955, the Laboratory grew into the "Institute for Mathematical Machines" and since 1958 held the name Research Institute for Mathematical Machines (well-known as VÚMS – Výzkumný ústav matematických strojů). In 1965, following the wish of the employees (cf. Jiřina, Korvasová), VÚMS became a research institute of ZPA, a Czechoslovak enterprise producing computers. In 1970s, around 1000 people were employed in VÚMS, of which over a third held a university degree.

At universities, computer science existed without computers – for example, computers were not allowed within the grounds of Brno University until the late 1960s. Until then, university researchers used computers (ZPA, LGP, Minsk, Ural, etc.) at the polytechnic. The limited access to technology might have (as a side effect) contributed to the rise of programming culture in Czechoslovakia.

In 1960, when the ALGOL 60 report was published, it represented a milestone in language design (cf. Young, p. 218). It is considered to be one of the most important, perhaps even the most important, language in history, as it introduced many revolutionary concepts that are used in programming up to now. In contrast to the development in the USA, dominated by Fortran and Cobol, ALGOL 60 did not have an equally appealing alternative, at least not in the area of scientific computing. It is within the scope of the project to study what helped the programming language ALGOL 60 to become the language No. 1 in Czechoslovakia by mid-1960s. There were local versions of ALGOL 60: e.g. ALGOL MSP 2a for the MSP2a computer, ALGOL GENIUS for music composition, or ALGOL ADT for the ADT computers. Additionally, ALGOL-based languages (especially PEARL and Pascal) appear to have been influential on generations of programmers.

Aims and objectives

While focusing on the software/computer science community in postwar Czechoslovakia, I plan to show how tradition (especially German/Austrian and French) mixed with Russian (Soviet) influence and how it created (central) European identity in this specific field of science. Long co-habitation with Germans under Habsburgs (1526/1620-1918) is e.g. still apparent in the school system. The short, but often recalled, period between the wars (1918-1939) is marked by strong French influence in the cultural/political sphere. Finally, the postwar period is marked by Russian/Soviet, and Communist influence together with the consequences of the resulting brain drain. The nature of software/ computer science allows for easy cross-border relations, since it is not bound with the specific context of a nation, although the Cold War made travelling abroad more complicated.

Did the programmers influence the society as a whole? Such an assumption comes naturally, considering the fact that software is an absolutely necessary prerequisite for most people to be able to communicate with (nowadays almost omnipresent) computers. The research aims at showing how the ALGOL culture in programming contributed to Europeanness.

Individual project 3

Using IBM in Europe to recapture the lead?
Co-constructing computer expertise in Europe and visions of European know-how through IBM and its technology

Hannu Salmi, University of Turku
with Petri Paju, Postdoc , Turku

Aims and objectives

This project aims to scrutinize the co-shaping of computer expertise and visions of European capabilities through the use of IBM technology in Europe during Cold War. IBM Corporation and technology had extensive influence in Europe as elsewhere in the era of mainframe computers. They are therefore essential in understanding both European experiences in computer use (incl. software) and the processes in which visions of European and national computing and data processing were planned, negotiated and performed. Instead of just adopting foreign technology and absorbing alien culture from outside, as has long been the dominant view on Europe's relation to the IBM, did Europeans reshape and reinterpret IBM technology to fit their own purposes? Further, this research offers new insights into the roles that multinational companies have played and can play in shaping Europe.

The use of IBM technology in shaping Europe is analysed in several levels of which two most important are the European (IBM Europe) and national (focusing on IBM Finland). In addition IBM's Nordic office will be examined in relation to the above mentioned levels. All of those are studied with respective users. Focus is on time period of 1950-1980. These levels ensure that the topic will be addressed in its complexity.

Framing Europe and negotiating its capabilities

IBM founded IBM World Trade Corporation subsidiary in 1948 to handle its business outside of the United States of America. This brought together IBM units of which some had been in existence since the 1920s in Europe. Arthur K. Watson, the second son of the famous Thomas Watson, joined IBM World Trade Corporation soon after its formation. In 1954 he was elected president of the IBM World Trade. According to IBM Archives web-pages, Arthur Watson early on supported the establishing of the European Economic Community. He wanted the IBM to be among the first U.S.-based companies to take advantage of the Common Market. Perhaps tellingly of his political interests, in 1970, Arthur Watson became U.S. Ambassador to France until 1972.

In Arthur Watson's command IBM Europe contributed to modernizing postwar Europe with new technology and applied science provided by this large and influential U.S.-based company. In this research project, instead of aiming at a full-scale corporate history of IBM Europe, we ask what kind of Europe did IBM Europe foresaw and construct, and how did they negotiate this policy with IBM Headquarters in the United States? How independent was the IBM Europe? Could it do research on its own or was the IBM (Corp.) mainly responsible for research and development work?

IBM Europe seems to have had a continental strategy for the emerging computer market in 1950s. In 1955, a well-known data processing and computing center with the first IBM 650 in Europe was opened to customer service in Paris. One could argue that the material technology itself, hardware and software were in the core in integrating European and other IBM producers and users. The same year an IBM 650 factory started in Stuttgart and a research center in Zürich. Several other computing and education centers were created in the 1960s. People from IBM national subsidiaries visited these for courses and met European colleagues and users. Strategically, IBM also penetrated universities offering computers with special discounts so that the students, future computer specialists, would learn to use IBM computers and software very early on. IBM managers traveled around in Europe. For instance Arthur Watson learned European languages and spoke French, Spanish, German and some Portuguese.

Although IBM had competitors in Europe, both national companies and U.S.-based companies like Remington and Burroughs, it was frequently in a position to choose with whom to cooperate with. For example, IBM turned down an early offer to work for what was to become the ALGOL programming language which is studied in detail in other Software for Europe individual projects. What kind of disintegrative practices did IBM follow in order to acquire a dominant market position in Europe? Still, were there European contributions in co-developing IBM's programming languages? Did these programming languages and program packages contribute to a hidden integration among the IBM communities inside Europe and the ones in the United States?

In 1955 an association called SHARE was founded for major IBM users to share information and give user feedback to the IBM. Some European users participated in the meetings early on. In 1966, SHARE European Association (SEAS) was incorporated as a non-profit making association. In 1994 SEAS decided to merge with another international user organization founded in 1959, and they formed a new association. Atsushi Akera (2001) has studied the SHARE in the U.S. but does not mention the SHARE Europe. In this collaboration and negotiating with IBM, what kind of visions and arguments of Europe were expressed and how did people experience being European and using IBM technology?

Re-building Europe consisting of nation states

It seems IBM's national subsidiaries were and perhaps needed to be both international (incl. European) and nationally mindful at the same time. For example, on the one hand IBM Finland followed instructions from IBM Europe and IBM Corporation headquarters. In computer service IBM Finland depended on IBM Europe and its sister subsidiary in Sweden and thus gave its customers a truly international or European experience of IBM know-how. On the other hand, in its board IBM Finland had notable Finnish politicians (incl. a foreign minister) and the company was eager to secure a nationally dominant position in the emerging computer market. Probably typically in Europe, Finnish data processing association (at first a punched card user association) started in 1953 as a reaction to the IBM Finland. Those IBM users wanted to create pressure for it to better its services. IBM Finland responded to this move by the users and become the market leader in computing as it began. This was mostly because IBM was perceived international leader in technology and business – but was it also because using IBM was a way to maintain close connections to the U.S. and West Europe in times when the Soviet Union pressed to opposite direction (Paju 2005, s. also studied by Durnová)? It is therefore worth asking, how did IBM's national subsidiaries construct national or European community or did they connect them into a process of co-construction?

With strong experience in military systems from the United States IBM provided many European countries with crucial military installations and military computer capabilities. IBM France designed the STRIDA air defense system in early 1960s, for example (Mounier-Kuhn 1998). In Finland the general headquarters became an IBM client in 1954. In the Cold War period this dependence must have made the national governments reluctant to hear criticisms of IBM – probably especially so in the NATO countries. Did this change in the 1960s when the "technological gap" between USA and Europe rouse to discussion and European governments began to foster "national champions" (in Finland: Nokia?) to compete in high technology markets? In the Finnish case, did tightening relations with the Soviet Union bring problems with IBM?

IBM in general appreciated marketing as a valuable asset and also spent a lot of money in publishing magazines both for internal company use and for the customers and wider dissemination. These journals spread information on the above mentioned activities and probably integrated the IBM communities on both sides of Atlantic. How was Europe on the one hand and nation states on the other represented in IBM journals? Which integrative strategies did this trade press use? Which topics were left out of publicity?

In these many levels and complex processes both IBM's roles in building Europe and European's multifaceted use of IBM technology will be studied to achieve better understanding of technology in the making of present-like Europe. Through these studies we could argue that using IBM technology did not mean that it had the same social impact or simple cultural meaning inside Europe and for European identities. Moreover, we could examine those impacts and meanings. In addition we will shed new light on the roles that multinational companies have played in constructing – integrating as well as disintegrating – postwar Europe.

Associated project 4

Competing Modernities: 'Machines Américaines', Fortran & Algol in Belgium, 1940-1968

Sandra Mols, University of Namur

Introduction: Aims, Objectives & Methods

In the early 1960s, Belgium, divided between Dutch-speaking Flanders and French-speaking Wallonia, was ill-equipped in computing equipment. There was no indigenous computing industry, no computer science department, and national science institutes had only recently been equipped with electronic computers. Belgian computing expertise was in fact imported from the Anglo-Saxon world, locally reshaped and appropriated, but an import nevertheless. This relative backwardness, also cast as a decline due to comparisons to Belgian science and technology in the interwar period, has been related to World War II, conservatism in politics, economics and science & technology, and the nation's dual linguistic structuring. (Deprit 1960; OECD; Halleux; Lambert). Similar observations apply to programming in the 1960s, which seems to have consisted in mimicking, with delay, American and British sophisticated programming practices. In this project, I explore these delays, backwardness and process of appropriation by (a) elaborating inventories of resources for the history of computing technology and computer practitioners in postwar Belgium, and (b) analyses of programming practices developed in Belgium, and their relations to the emergence of a Belgian computer industry and of 'informatique'/'informatica' university departments from the late 1960s onwards.

Belgian Computing Technology and Practices, Unexplored Historical Problem

Belgium emerged from World War II severely disrupted, materially, economically and socially, its politicians facing the complex task of the reconstructions of its economy, infrastructures and even society as World War II had contributed to reinforce Flemish nationalistic movements. The modernisation of science and technology was planned in this reconstruction.

The IRSIA-FNRS Computer

Trips to the US and UK and reports on Anglo-Saxon advancements in science and technology, and electronic computing research (Brillouin; Halleux), as in the Netherlands, inspired in 1951 the Fonds National de la Recherche Scientifique and the Institut pour la Recherche Scientifique dans l'Industrie et l'Agriculture to sponsor the construction of a national computer, the 'machine IRSIA-FNRS', using American designs. The machine was completed during 1955-1956 and scrapped in 1962. The 'Comité d'Etude et d'Exploitation des Calculateurs Electroniques', constituted of delegates from the military, elite physical sciences academia (Dutch and French-speaking), and technicians from the Antwerp branch of the American Bell Telephone Manufacturing Co, was in charge of the project. (CECE). Against expectations, instead of a boosting up of home skills in computing technology, the main impact of this national machine seems to have been a brain drain of the few Belgians expert in electronic computing towards foreign firms, e.g. Bell, or the Dutch company Philips. Also, the machine did not generate indigenous know-how in computer use for being mostly used national and military science projects. All in all, feelings of backwardness seem in fact to have reinforced until the turn of the 1960s, due to constant news of, and exposures to, American and British computers.

Academia-Industry-National Science Networks

Until the late 1950s, universities seem to have remained deprived of modern computing facilities. For instance, at the Université Catholique de Louvain, Leuven, until 1958, priorities lay with recovering from the war and computing facilities remained limited to prewar electro-mechanical desk calculators housed at the Laboratoire de Recherches Numériques (LRN). In 1958, the university purchased an Elliott E101. Criticisms were however soon expressed about the comparative non-modernity of this new equipment, due to words of mouth spreading about IBM, Burroughs and Ferranti computers seen e.g. at the 1958

Brussels World Fair. It was only at the turn of the 1960s that the University established an American-styled 'Centre de Calcul' equipped with IBM equipment, a unitary facility later removed to the franchised French-speaking Université Catholique de Louvain, relocated in 1969 to Wallonia. Collaboration with IBM then developed at the new Dutch-speaking Katholieke Universiteit te Leuven (Leuven IBM Education Centre; LUDIT). (Florizoone; Woitrin; Mols; Lambert). Information is very limited about other institutions. Interviews suggest little: electro-mechanical calculators in Liège, CDC computers in Brussels, a mid-1960s project by French-speaking Belgian universities of an interuniversity computing facility about which it is unknown whether Dutch-speaking universities were involved, which equipment was chosen, and whether it materialised. Establishing an inventory of computing equipment in use towards Belgian academic research could prove significant to a history of Belgian academia. Such an inventory could explain the institutional inexistence of Belgian computer science until the 1970s, and the ways in which the French-vs.-Dutch dichotomy inherent to postwar Belgium affected developments in computing. Another major issue requiring exploration is also the extent to which networking with industries (Agfa-Gevaerts), national science institutes (e.g. the Institut National de Statistique) and abroad (e.g. Amsterdam, York, CERN, University of Berkeley) helped Belgian postwar academics compensate for their lack of equipment. Unravelling these networks – which were at first glance quite extensive - would contribute to assess the reality behind backwardness complaints.

The Late Emergence of Modern Programming in Belgium

As regards to programming, the information currently collected suggests that Belgian practices was a mix of Anglo-Saxon (Autocodes, Fortran) and more typically 'European' (if ALGOL can be cast as such) approaches. Like in 1950s Britain, first, the kind of machines in use seems to have dictated the kind of programming language and techniques in use. E.g. the presence of the FNRS-IRSIA machine seems to have led to programming groups at the University of Antwerp; at the LRN, the purchase of an American Burroughs E802 inspired the writing of autocodes; IBM machines came hand in hand with a popularisation of Fortran and IBM operating systems. (Mols; Lambert; David *et al.*; Florizoone). The impacts of such factors as machine technicalities, business policies, and pre-existing programming habits in this appropriation of Anglo-Saxon savoir-faire is yet to be assessed however. Did machines really dictate programming practices?

As for the more pure mathematics-based ALGOL, preliminary investigation suggests, first, a primary spread during the early 1960s. In Louvain, in Bruxelles, a word of mouth spread about the ALGOL project, via, mainly, Manneback, internationally recognised physicist (1894-1975), and, possibly, De Vogelaere, computer scientist at the University of Berkeley (Deprit 1994; Lambert; Chambre *et al.*). This import seems motivated by the opinion that ALGOL was a better, more modern, language due to its more mathematical origins. Second, ALGOL was re-imported from 1964-1965 onwards from (a) the Grenoble Institut de Mathématiques Appliquées, France, and the Zürich Eidgenössische Technische Hochschule, both pioneering places as regards to the ALGOL effort, Switzerland (ETH); and, a bit later it seems, at the turn of the 1970s, from (b) the Netherlands via Philips, van Wijngaarden, and the Manufacture Belge de Lampes Electriques (Wijngaarden *et al.*; Magis). As regards to ALGOL, research questions aim in particular at exploring whether the ALGOL enterprise was (a) the epistemic consequence of European computer scientists' taste for more mathematics-based, 'purer', programming, an hypothesis suggested by the first importation of ALGOL in Belgium, or/and rather (b) a European counter-reaction against the growing dominance of IBM during the 1960s. In a few words, was ALGOL a computer science achievement or about Cold War/EEC geopolitics? Or both?

Associated project 5

Software Development Configurations: The 'IMAG' Lab at Grenoble and the Computer Industry

Pierre-E. Mounier-Kuhn, Université Paris-Sorbonne

The French academic scene has traditionally been described as a highly centralized system dominated by pure science values[1]. This description is partially true, as it reflects mainly the point of view of pure scientists living in Paris, who were for decades the primary source of information for whoever studied the history and sociology of French science. Yet the picture differs significantly if one changes focus and point of view. If we switch to considering applied sciences and provincial universities, we discover, stemming from the 1860s, a landscape of relatively autonomous faculties which often fostered technical activities to respond to practical, local needs, sometimes in competition between State and private (generally catholic) higher education[2]. In cities like Toulouse, Grenoble, Lille and elsewhere, since the late 19th century, the Facultés des Sciences raised funds from local industries and city councils; they developed engineering departments based on the German model of the *Technische Hochschule* (and on the Belgian model of the Institut Electrotechnique Montefiore at Liège, in the case of Nancy); they created chairs and laboratories in disciplines such as chemistry, hydraulics, electricity, later aerodynamics and defense-related fields. In 1913, the Facultés des Sciences in Grenoble, Toulouse and Nancy produced more graduates in technology than in 'science' *stricto sensu*. Around 1950, the same universities hosted large engineering schools and laboratories in these fields, where the need for applied mathematics and computing means was becoming important[3]. Contrary to many University establishments in other industrial countries, the French academics did not build store-program computers in the pioneer era[4]; they focussed initially on the use of computing machines, initially on numerical analysis, then progressively on developing various aids to computer use (compilers, languages, program libraries, etc.), operating systems, and theories to help understanding and advancing the computing technologies. In other words, old traditions of industrial-academic cooperation, and configurations of regional interests and international models, created conditions favourable to the emergence of software — of programming practice, R&D, and training in this new art[5].

The early computer scene, particularly in Europe, was characterizd by the scarcity of human resources. To overcome this problem, academic groups and computer manufacturers joined efforts to develop the most novel product: software — particularly basic software. My project aims at understanding how academics and industry engineers built co-entrepreneurship and experimented new ways for further cooperative research. The case studied will focus on the relationships between the Grenoble 'IMAG'

---

1 Among many publications on this classic theme, see for example Gilpin R., *France in the age of the scientific state*, Princeton University Press, 1968.

2 H. W. Paul "Apollo Courts the Vulcains - The Applied Science Institutes in Nineteenth Century French Science Faculties" in R. Fox et G. Weisz, *The Organization of Science and Technology in France*, *1808-1914,* MSH Cambridge UP et MSH, Paris 1980 ; G. Weisz *The Emergence of Modern Universities in France*, *1863-1914,* Princeton UP, Princeton 1983. Grelon, "Les universités et la formation des ingénieurs en France (1870-1914)", in *Formation emploi*, n°27 -28; & M. J. Nye *Science in the Provinces : Scientific Communities and Provincial Leadership in France, 1860-1930*, University of California Press, 1986.

3 Grossetti M. & P. Mounier-Kuhn "Les débuts de l'informatique dans les universités - Un moment de la différenciation géographique des pôles scientifiques français" *Revue Française de Sociologie*, 1995-2.

4 An overview of this story was first presented at the *Technological Change* conference (Oxford University, September 1993): P. Mounier-Kuhn, "The Failure of the French Public Research System in Building Pioneer Computers". A more comprehensive version was published later: P. Mounier-Kuhn, "L'enseignement supérieur, la recherche mathématique et la construction de calculateurs en France (1920-1970)", in *Des ingénieurs pour la Lorraine - Enseignements industriels et formations technico-scientifiques supérieures, XIXe-XXe siècles* (F. Birck & A. Grelon, dir.), Editions Serpenoise, 1998.

5 Mounier-Kuhn P., *Le Comité national et l'émergence de nouvelles disciplines au CNRS : le cas de l'informatique (1946-1976*), Mémoire de DEA, Centre Science, Technologie & Société, CNAM, Paris 1987.

(Institut de Mathématiques Appliquées de Grenoble) and the computer companies operating in France — both being elements of European scientific and industrial configurations.

At the present stage of my research, the overall picture can be summarized as follows. IMAG scientists' participation in the ALGOL group altogether confirmed Grenoble's role as one of the stars in the European software science galaxy, and paved the way toward further cooperative work with the industry. First, Grenoble computer scientists wrote compilers and other software tools under contract for most French computer manufacturers. A second, deeper relationship was established around 1970, when IBM, followed by the national champion CII, created “Scientific Centers” in the IMAG premises, mixing academics and industry software developers, to undertake joint research on novel concepts (virtual memory, network analysis and modelling). In turn, this collaboration induced a new research culture within the computer industry, and contributed to legitimize the emergence of “basic software” as a distinct function and structure within companies like Bull and CII. These efforts were transnational from the beginning — for example one of the CII engineers at the Grenoble Center was an engineer from the Delft Technology University (Holland), Harm Smit, who had worked with Philips Data Systems before joining the French Plan Calcul. In the mid-1970s, the CII-IMAG Center in Grenoble developed software tools and network models for the next generation of Unidata computers, designed in the European joint venture created by CII, Siemens and Philips.

Further research is necessary to validate these hypotheses; to grasp the circulation and transfers of models (organizational and technical as well); to analyze the development of a new software culture bonding together universities and companies in Europe (West and Est) and in North America; and to understand how the interactions between these forces, together with national and European technology policies, shaped innovative capabilities.

A permanent element of this story is that, in the period considered, software meant tensions :

- Tension, in the University, between pure science interests and cooperation with external partners (industrial or military), which led to a fierce controversy, from the mid-1960s on, about the nature of software (craft, or science?), and therefore of its statute in academic organizations.

- Tension, on the human resource market, between the small number of programmers and the fast growing demand from computer developers and users.

- Tension, as was acutely expressed in the *software crisis* debate, between the progress of computer hardware, the ambitions and promises of marketing, the demand of users and the actual programming art & craft of the late 1960s.

- Tension, in the professional circles, about who should be entrusted with the development of operating systems for the national champion CII's computers : Should basic software be developed under contract by the software and service companies (SEMA was an obvious candidate, as it employed high level experts who actually developed time-sharing systems for Control Data and other manufacturers)? Should it be entrusted to academic labs like IMAG, CERA or IRIA ? Or should the small CII itself develop the whole systems, while it was already struggling with an overambitious product plan ?

The present research project will investigate these tensions and how they interacted, the way they were perceived, the compromises which were struck to solve them.

Associated project 6

Software in Britain — computer appropriation, automatic coding and the problematics of the 'British Problem,' 1948-1970

James Sumner, University of Manchester
with postdoc NN, Manchester

Consideration of British institutions and actors is essential in fully understanding the heterogeneous character of European software development. 'Software in Britain' will use the case of the United Kingdom to examine the creation, appropriation and representation of computer software in Europe between the late 1940s and 1970.

Aims and objectives

Besides maintaining one of the world's largest data-processing industries, the United Kingdom developed, at both policy and technical levels, distinctive systems and approaches rooted in a broader cultural exceptionalism. Postwar, post-imperial Britain — anglophone, enthusiastic for Nato, geographically and culturally removed from the heart of Europe, and initially sceptical towards European union — was widely perceived to share a stronger common context with the USA than with its immediate neighbours. Yet the concept of the US client state implied decline and marginalisation, and was for that reason widely resisted by policymakers.

When schemes for international integration and standardisation came to be seen as 'European' or 'American,' questions of acceptance and rejection became politically and culturally loaded; some in the field were keen to invoke an altogether independent technical and cultural identity for the United Kingdom. These tensions are strikingly exhibited by the case of electronic information-processing.

In the late 1940s Britain, principally through the academic-industrial complexes of Manchester and Cambridge, was unique in attaining joint leadership with the USA in storage, processing and programming (Campbell-Kelly 1982; Campbell-Kelly and Aspray 2004), with enduring legacies in building computer-science institutions and training researchers and coders. Many continental European researchers, indeed, considered research visits to Britain as a 'stepping stone' to American work. In commercial manufacture, however, the old country was swiftly eclipsed by American corporate expansion, most conspicuously on the part of the dominant International Business Machines (IBM.)

This story, familiar from John Hendry's *Innovating for Failure* (1989), is commonly considered as a classic case of the 'British problem': the supposed national pathology of faulty priorities, fostering superlative technical innovation which is then routinely defeated in a commercial environment. Initial discussions between the Software for Europe partners, however, suggest that there is nothing specifically British about the problem: similar stories are told in France, the Netherlands and elsewhere. At a deeper level, moreover, the declinist tendency in writing the history of British industry has been queried. David Edgerton (1996, 2006) presents the 'decline' narrative as a discursive strategy exploited by various constituencies (in particular, the military) to benefit their interests.

The proposed project will expand this problematisation through a focus on the 'hidden' technology of software: if national presence in computing activity can be gauged from the use and users of computers, rather than from production and distribution of the machines themselves, the declinist narrative may be challenged: the 'British problem' (along with its analogues) may itself be problematised.

Software in Britain reflects the overall SOFT-EU focus on tensions between the artefact and concept modes of appropriation, looking both at IBM's entry into the UK and British academia's relationship to the ALGOL programming-language initiative. The former investigation will usefully counterpoint established British literature (Hendry 1989; Campbell-Kelly 1989) in focusing on imported rather than indigenous computing cultures. The UK was the only European nation in which IBM's task was to displace multiple established firms: whereas, elsewhere, IBM's multinational character could lend itself to pan-European rhetoric (cf. Paju's proposal on the Finnish case), the anglophone UK was more prone to perceive US expansion and local decline.

The second research strand, accounting for two-thirds of the projected output, is to focus on Britain's role in the evolution of programming languages. The problematics of the supposed polarisation between the 'commercial,' IBM-originated FORTRAN language and the academic 'purity' of the rival ALGOL standard is addressed at length by other Software for Europe partners. To these the British case adds additional complexities. Programming languages (known initially as *autocodes*) were pioneered on working machines in Manchester and Cambridge from the early 1950s, predating both FORTRAN and ALGOL in application. This early emergence of a technical base left British researchers choosing between established resources (depth) and cross-border appeal (breadth) when international standards emerged later on.

Expertise in programming, and in the provision of the means for programming, crossed national and cultural boundaries more easily (and less visibly) than did the results of export-order contracts for information-processing machinery. ALGOL, in the long term, failed as a standard at the level of practical use; the survival of the assumptions and priorities embodied in its development, however, suggests it may simultaneously be considered a success at the conceptual level.

The project overall, then, will go beyond simple 'competition' narratives, to address relations between organisations (whether commercial or, as in the case of UK universities, primarily state-funded) which might involve assimilation, co-operation, competition on common ground, or rejection in favour of independent alternatives.

Research questions

Was there a "European space for software," or a European mode of software development, in the period 1948-70? What was the role of European identity — and of its alternatives, national and international — in determining software policy in Britain and elsewhere?

Why did IBM (and, to a lesser extent, its US rivals) overcome the established strength British computer industry so thoroughly, and so perceptibly, before 1970? Was this simply a case of economic might, or were more subtle social, political and technical factors involved?

What part did Britain, with its established automatic-coding tradition, play in endeavours such as the ALGOL effort? How far did the mathematically-oriented 'purity' culture resonate with British researchers? Where resistance was felt, what was its constituency, and how did it manifest itself?

To what extent does software offer a means for 'hidden integration' across national, social, political, economic and cultural boundaries?

What, finally, is the significance of the 'British problem' in European context? To what extent is it valid? What does the British problem's popularity as a fable reveal about British and European projections of technical identity?

Methodology

The research will be qualitative in nature, informed by the disciplinary traditions of science and technology studies and the history of technology. Documentary research, mostly in institutional archives, will be used to guide an extensive programme of interview work. Established literature on the writing of oral history commonly focuses on the stories of 'ordinary people,' as a valuable corrective to the archives' tendency to favour the evidently powerful (Thompson 2000.) In cases which have not been long perceived as historically significant, however, the archival record may be incomplete or unassembled; there is thus an emerging parallel tradition of 'elite' oral history, whose techniques inform our work. Besides interviewing principal policymakers, developers and representative users, we will address recent appeals to consider non-users (Wyatt 2003) by including those who made specific decisions *against* the relevant appropriations.

Interpreting software developers and users as a community with a variety of shared and conflicting imperatives, the project will employ the well-established technique (Latour and Woolgar 1979) of bringing ethnographic insights to bear on a technologically-advanced expert group. Information drawn from interview work will in turn be compared against surviving literature and correspondence from the period, giving the widest possible understanding of the independent development, acquisition, modification and re-presentation of software technologies. It is anticipated that this process will include examination of software code itself, a hitherto under-used resource which may yield invaluable insights into programmers' priorities and intentions.

Associated project 7

Software Tensions in Non-Anglophone European Contexts: The Example of Greece

Aristotelis Tympas, National and Kapodistrian University of Athens
Theodore Lekkas and Dimitris Ziakkas (Ph.D candidates), Athens

Aims and objectives

This project focuses on the history of the tensions caused each time citizens of several European countries realized that software imported from the US could not support applications requiring the use of their own alphabet. Given that post-World War II European integration initiatives started by guaranteeing the diversity of European languages and the cultural traditions associated with them, the difficulty to have software that could technically sustain this linguistic diversity became a chief European technological problem, one that stood in the way of all attempts at europeanization. It became a problem of vital importance for some of Europe's smaller countries, which felt that they could not address the issue at the national level only, because of the lack of proper economies of scale. To study the details of this problem, we will focus on the case of Greece.

The problem kept surfacing each time the use of computers was extended. It was felt most acutely when it was realized that imported software, despite being presented as 'universal' ('general purpose'), could not support most word processing uses. According to representatives of software imported from the US, the issue ought to be simply one of proper installation of the right language features, already available in the imported software package. At the other end of the spectrum, Greek software houses were developed to remedy the issue, based on the argument that the problem was not simply one of proper installation. On the contrary, following the complaints of the majority of the users, they suggested that a major reprogramming effort, going deep into the software and even requiring modifications of the programming language employed, was indispensable. Noticeably, it was not simply a problem of improper appearance of the Greek characters on the computer screen. Users were constantly complaining about the inability of, for example, printers to produce proper texts in Greek. The problem went back to the age of the mainframes and survived during the age of the Web and the Internet.

The set of the research questions to be addressed by this AP includes the following: What were the technical solutions tried (or rejected), successfully or unsuccessfully, in response to the various manifestations of this problem? How exactly various europeanization initiatives (e.g. EC and EU software research projects) sought to solve the problem? How did such initiatives emerge and how did they score? How did such initiatives interact with national ones? Have there been international alliances (within Europe, and within European nation-states) in favor of certain technical agendas? Was there a difference in the response of public authorities and private firms? What was the role of computer science and technology academicians, professionals, and user groups? Was there a process of hidden European integration, due to success or failure, in dealing with this software problem?

Methodology

We plan to focus on common European initiatives (e.g. research programs), designed and developed as a remedy to this problem. We additionally propose to conduct research on the interaction between a nexus of European and Greek actors involved: US and European hardware and software vendor representatives, Greek and other European software houses specializing in solving proper language support projects, relevant clubs and associations of Greek and other European users, networks of Greek and other Europeans that have been circulating commercial and *ad hoc* software solutions to this problem (including piracy advocates and opponents), technician groups and clubs, academicians and their proposals (there seems to have been a tension between those trained in the US and those trained in non-Anglophone European environments), Greek standardizing committees (usually controlled by the state), software specialists offering advices through their role as columnists in Greek software journals, ideologues rejecting or supporting the development of computing in Greece based on their view of the software problem. Our primary sources are computing journals, European Union archives, and interviews with protagonists.

Associated project 8

ICT and Business. The Rise and Development of Sofware-based Industries in Europe (Integration – Homogenization – Differentiation)

Paul Erker, Deutsches Museum, Munich
Timo Leimbach (postdoc)

The spread of Information and Communication Technologies (ICT) since the 1960s has changed our patterns of public life as well as of private life. The cutting edge of this fundamental change was the implementation of these technologies in corporations. „Since the 1950s business has been as radically transformed by the digital hand of information technology as it was by the managerial revolution in the period Chandler described". This is the main argument of James Cortada, one of the very first historians who tried to describe and analyze this development in American business. In most cases, the implementation of ICT (Hard- and Software) was motivated by two aims: Firstly, management aimed at getting a better financial and managerial control of the corporation and secondly, companies used ICT to rationalize and optimize the production process, the material and information flows and related factors. Because of the fact that both, white-collar as well as blue-collar work was concerned of this, it fundamentally changed the structure, organisation and policy of corporations. On one hand, totally new industries like consulting or online retail were created and on the other hand, traditional industries like manufacturing enforced their innovation life cycles with the adaptation of ICT. The transformation process of both, "new" and "old" industries varied in range and scope.

While the history of producers of ICT like the computer hardware is well understood, the "demand and consumer side" of ICT, especially the effects and consequences of its use, has not been examined for the European market so far. While for the U.S., the most recent works of James Cortada (2003, 2005) and JoAnne Yates (2005) have contributed to a deeper understanding of this ongoing process, for Europe the lack of studies is all the more crucial. Thus, the focus of this project is to analyze this process for the European corporations and markets.

Aims and objectives

As for the conceptual design of the project, research on the implementation and use of software in corporations will be the most important, but also the most challenging part of the work. The reason of the significance of software is that while the costs of hardware have decreased rapidly and hardware has become a standardized and compatible good since the 1970s, the cost of developing and especially of implementing software have increased and caused a lot of corporate problems. Due to these problems, experts, producers and users started to talk of a "software crisis" or a "software application bottleneck".

Yet, the aim of the project is to examine the intersection between technology, business and economy. Four major, interdependent aspects of research can be identified:

1. The process of implementation and use of ICT, especially software, concerned all spheres of business; it stretched from the production of consumer and producer goods to the provision of services, internal communication and customer relationships, finally to decision making and company organization. All these spheres and lines of this development should be traced back and analysed in all commonalities and particularities. Exemplary industries to be investigated will be mechanical engineering, transport, finance and banking, heavy industries (steel and iron industry/mining), suppliers (especially power suppliers), automotive industry and chemical industry. An international comparative approach between Europe and the U.S. will be used to answer two leading questions: Firstly, what are the differences between various industries, and secondly, is there a European way of implementation and use?
2. Due to the fact, that the adaptation of ICT and software marked a point of no return for companies, the change of technologies developed into a crucial avenue for success or failure for corporate actors. Decisions on the implementation and use of new technologies were conjunct with high financial and strategic risks. In the worst case "wrong" decisions resulted in essential corporate crisis.

3. Two aspects of this overall process of change have to be specified more precisely; Firstly, the necessary customizations which emerged from the use of software in companies; secondly, the increasing needs of users towards the further development of hard- and software. As a consequence, the question of synchrony and asynchrony of technological innovations in ICT industry and their adaptation in other industries has to be tackled. To tackle this question will contribute to the central debate on the impact of the use of ICT on corporate productivity which started in the late 1980s with the discussion on the so called “productivity paradoxon”. Most recently, Nicolas Carrs has contributed to this ongoing debate with his assumption that “IT doesn’t matter”.
4. Due to the fact that the spread of hard- and was mostly precipitated by IBM, which creates a close connection to the other projects in this CRP, and the emerging software firms, products and market in the late 1960s, enabled through IBM’s Unbundling and the rise of tools like high-level programming languages, it is necessary to use this developments as a starting point to understand the technical needs and problems of the implementation of ICT and also to track institutional and personal networks between software firms, consultants and corporations and their changes, which are involved in this course of activities.

Associated project 9

Genesis and Development of “Soft Computing / Computational Intelligence” in the 20th Century European System of Science and Technology

PD Dr. Rudolf Seising, Munich

This project is an investigation of the genesis and development of “Soft Computing / Computational Intelligence” research in Europe in the second half of the 20th century as an example of “mode 2”, which has been proposed as an adequate approach to model knowledge production in modern knowledge societies.[6]

The project discusses important contributions to the shaping of computer sciences and artificial intelligence in Europe after the Second World War. It is associated with the collaborative European project “Software for Europe” (Eurocores) even though it does not address any specific software development.[7]

To this end, we must consider developments in humanities, science, and engineering in the first half of the 20th century, as well as the origins of the theories and methods that constitute the field of “Soft Computing / Computational Intelligence”. We must go back to the beginnings of computers, the theory of the Turing machine and of automata, and to the mathematical formulation of bionics and evolutionary strategies as well as to the analysis of natural neural networks and the construction of artificial neural networks.

In research on artificial intelligence in the last third of the 20th century, some scientific theories and methodologies came into being that differ in essential features from older theories and methodologies in science and technology. These theories and methodologies merged to form the field of “soft computing” – a name coined by Lotfi Zadeh, the founder of the theory of fuzzy sets (1965) and “possibility theory” (1978).

In 1991 the term “soft computing” was first put forward in contrast to the “hard computing” of exact data processing. Whereas the latter uses precise algorithms to derive conclusions or optimal values from clearly defined data, soft computing takes into consideration concepts that are not clearly defined, inaccuracies, vagueness, and blurred knowledge. Other segments of the field of soft computing are artificial neural networks, evolutionary and genetic algorithms, and probabilistic reasoning.

---

6 Michael Gibbons, Camille Limoges, Helga Nowotny, Simon Schwartzman, Peter Scott und Martin Trow: *The New Production of Knowledge. The Dynamics of Science and Research*. London 1994 (Sage).

7 Gerard Alberts: Proposal for a Collaborative Research Project in Eurocores’ Software for Europe.

In hard computing rigid rule-based programs are used for computing and simulating logical reasoning, strategies, and decisions. Soft computing, on the other hand, targets “pre-intelligent” abilities of flexible and fault-tolerant perception and reaction made possible by the human sense organs and locomotor system – though technical solutions do not have to imitate natural solutions. The areas of soft computing are also known as parts of “computational intelligence”. This denotation was proposed by James C. Bezdek, a computer scientist in the USA who was another fuzzy set theory pioneer. He enumerated the following properties of soft computing methods, namely flexibility, fault-tolerance, velocity of processing in the area of human cognitive processes (particularly using inherent parallelism), and optimal error rates (proportion of learning expense to fault frequency).

With the adjective “computational” Bezdek referred to subsymbolic (numerical) problem representation and the consequent subsymbolic knowledge aggregation and information processing. The concept of computational intelligence, which emphasizes the calculative aspect in opposition to a symbolic approach, came from research circles focusing on artificial intelligence, whereas the concept of soft computing grew out of work done on fuzzy set theory, in which the continuous is accentuated in contrast to the dichotomy of traditional modelling.

The main areas of soft computing /computational intelligence came into existence – largely independent of each other – in the second half of the 20th century. They have the common property of imitating structures or behaviour in nature in order to optimize certain processes of problem solving where problems either cannot be resolved with the help of classical mathematics or can only be solved with great difficulty (e.g. fuzzy sets and probabilistic reasoning simulate human reasoning and communication, artificial neural networks are geared to the structure of living brains, or genetic algorithms and evolutionary programming are geared to biological evolution).

The theory of fuzzy sets appeared in the centre of the field of “Soft Computing / Computational Intelligence”, booming in the 1980s and 1990s in Japan, and later in the western world. The second discipline is the research program on artificial neural networks that developed in the 1950s and 1960s. This discipline suffered a setback in the late 1960s and therefore did not play an important role until the latter half of the 1980s in research on artificial intelligence (AI). After this field experienced a renaissance in the 1990s, artificial neural networks have now become the main discipline in AI and “Soft Computing / Computational Intelligence”.

The most modern subfield of “Soft Computing / Computational Intelligence” is “probabilistic reasoning”, originating in mathematical decision theory on the one hand and in AI on the other. This field comprises theories and methods for the propagation of probabilities in causal networks with an updating mechanism as inference procedure. Here, we have Bayesian networks, Dempster-Shafer theory, certainty factors, und belief functions. More recently, chaos theory, learning theory, and the field of genetic or evolutionary algorithms have been subsumed under “probabilistic reasoning”.

This broad range of subject matter should be deepened by comparative case studies:

- A comparison of the development of the theory of fuzzy sets in the German Democratic Republic (GDR) and in the Federal Republic of Germany
(Manfred Peschel and the “GDR Fuzzy Working Group”, the “fuzzy initiatives” (“Fuzzy-Initiativen”) in the Federal Republic, particularly in Dortmund, Aachen, Braunschweig, and Magdeburg)

- The development of evolutionary algorithms in the Federal Republic of Germany (I. Rechenberg, H.-P. Schwefel)

- The assimilation of artificial neural networks in Europe

- Further developments from the theory of fuzzy sets to “possibility theory” in France in the 1980s and 1990s (A. Kauffmann, D. Dubois, H. Prade)

- The continuative approach from “belief functions” to the “transferable belief model” in Belgium (Ph. Smets)

- Projects on the theory of fuzzy sets and its applications in Austria (E.-P. Klement)

- Additionally, there have been “Soft Computing / Computational Intelligence” initiatives in nearly all other European countries. In the context of these developments, the *International Journal of Fuzzy Sets and Systems* and the first fuzzy conferences (EUFIT, EUSFLAT) as well as the first scientific fuzzy communities and the first *European Soft Computing Centre* in January 2006 in Oviedo, Spain, were established.

In 1992 congresses took place in the USA, Japan and Europe (in Aachen, Germany) in which representatives from these three disciplines came together for the first time. Since then, there has been close interdisciplinary cooperation and communication under the generic concepts of "soft computing" and "computationally intelligence".

Every four years the IEEE Word Congress on Computational Intelligence (WCCI) has been held and will continue to take place – 1994 in Orlando, 1998 in Anchorage, 2002 in Honolulu, and 2006 in Vancouver. This congress consists of three parallel organized session-tracks based on the following three conferences (organized separately in previous years): IEEE International Joint Conference on Neural Networks (IJNN), IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), IEEE International Congress on Evolutionary Computing (CEC).

In October 1990 the Department of Electrical Engineering and Computer Sciences at the University of California at Berkeley decided to found the Berkeley Initiative in Soft Computing (BISC) on March 13, 1991, during the ILP (Industrial Liaison Program) Conference in Berkeley. Since that time BISC has developed into a worldwide scientific community of individual researchers and organizations, connected via the Internet, who have a shared interest in soft computing / computational intelligence and its applications. (http://www-bisc.cs.berkeley.edu/BISCProgram/History.htm)

In this project the development of "Soft Computing / Computational Intelligence" is regarded as an example of mode 2 knowledge production in the European knowledge society of the 20th century. It shows interdisciplinary research, heterogeneous networking, and co-evolutionary processes between theory and practice, and cooperation between university and industrial research and teaching. Embedding the genesis and development of "Soft Computing / Computational Intelligence" in the organizational principle of mode 2 knowledge production seems appropriate, especially as publicity and media skills have played a highly visible role in this process of development and there is no clear line of separation between the producers and users of knowledge – thus the distribution of knowledge in society is blurred.

Associated project 10

Inside the Box: A History of the Software Package

Thomas Haigh, University of Wisconsin—Milwaukee

During the coming years I plan to conclude an ongoing examination of the social construction of the software package. Packaged software in its most literal form, a shrink-wrapped box containing disks and an instruction manual, was a taken-for-granted part of the computing experience during the 1980s and early 1990s. But the seeming naturalness of packaged software masks a complex history and a great deal of commercial, cultural and legal work done to transform computer code into a consumer product. Software has been produced and consumed since the 1950s, but usually in different forms. Indeed, recent shifts toward downloaded software and subscription plans have eroded the once dominant position of shrink-wrapped software.

Packaged software relies on a legal framework in which the rights of producers are protected, on the acceptance of banks and investors that packaged software companies work in a profitable industry, on the willingness of accountants to value packages on a company's balance sheet, on the willingness of customers to purchase something that may contain bugs they are unable to fix, and on the creation of a set of shared cultural understandings governing concepts such as the difference between a bug fix (free) and an upgrade (usually paid for), the issuing of regular updates, and the period of free technical support to which a purchaser might be entitled.

My project is producing an ongoing stream of scholarly publications and will culminate in the production of a monograph on the history of the software package from its origins in the 1950s to the present day. I am looking forward to pooling my expertise with that of the other contributors to the Software for Europe initiative, and of incorporating their findings into my own work.

Four sections of the project are already underway or completed.

1. **The first attempts to standardize and package complex programs for use at different computer sites**. This practice actually predates the concept of "software," and is much older than the commercial software industry. I trace the creation of the first packages to efforts conducted within the IBM mainframe user group SHARE, Inc. from 1954 onward to eliminate duplicated effort by establishing a library of shared programs. The role of IBM and its original user group in this section makes it an excellent fit with many of the components of the Software for Europe initiative.
2. **The origin of the concept of software**. This deals with original appearance of the concept of software in the 1960s, the difference between "software" and "program," shifts in the meaning of the term over time, and the gradual separation of software and services. It focuses particular on the user view, that is the perspective of the corporate data processing managers for whom packaged software was an alternative to the established method of in-house creation of programs as needed.

The only scholarly work to date on the history of the software industry is Martin Campbell Kelly's comprehensive "From Airline Reservations to Sonic the Hedgehog: A History of the Software Industry" (MIT, 2003), an economic history of the emergence of an industry with business history summaries of the stories of representative firms. My own work focuses more on the software package as a social artifact, on the role of cultural and legal institutions in shaping and supporting its development, and on the active work users in negotiating a viable framework for the use of packaged software.

3. **The construction of packaged software as a commercial good**. The first companies to specialize in the sale of packaged application and systems software to end-users were created around 1970. This mainframe software industry predated the higher-profile personal computer software industry. My analysis focuses on the industry trade association ADAPSO. ADAPSO provides a window through which to observe the work necessary to establish the software package as an artifact with a sound economic and legal basis for existence, including the extension of copyrights to software, the first software patents, and the earliest software licensing agreements. This work was supported by a Software History Center fellowship.
4. **A case study of packaged mathematical software.** Through a detailed examination of the practices of mathematical software creation and use from the 1950s to the present day I illustrate a number of shifts in the relationship between producers and consumers of mathematical routines. I argue that the history of mathematical software has been influenced by changes in computer technology, by the desire of a new community of specialists to legitimate themselves as respectable mathematicians, and by the acceptance within this community of an unfamiliar blending of commercial development, academic publication and quasi-open source software distribution. I'm particularly interested in the ability of mathematical routines to "black box" obscure mathematical knowledge and to transfer embedded practices between scientific sites. This work has taken place in conjunction with the Society for Industrial and Applied Mathematics and a major grant from the US Department of Energy.

Archival research on 1-4 is complete, and a number of publications have already been made. Over the next two years I will focus on three further areas to complete the project, before integrating all the material into a monograph.

5. **The modern open source software movement.** By reinterpreting the history of open source software in the light of its continuing relationship to practices dating back to the SHARE efforts of the 1950s I believe I can provide new insights into the special features of today's movement.
6. **Personal computer software and the shrink-wrap model**. The creation of a new kind of software industry in which software was packaged in colorful boxes, sold through retail channels, and purchased by millions of ordinary people rather than by thousands of computer center managers.
7. **Downloaded software and the Internet**. In the past ten years our idea of the software package has shifted again, as the Internet shifted practices toward the downloading of software, the

constant streaming of updates and what is often called “software as a service” (systems such as Gmail or Flickr where the application itself is accessed via a web browser).

Associated project 11, Yost

Software for Europe: History of the Computer Services Industry

Jeffrey R. Yost, Charles Babbage Institute, University of Minnesota

This project will examine the increasingly influential, but largely unstudied computer services industry. While the focus will be on the United States, understanding international developments and contexts of the trade will be fundamental to the project. Research and analysis of this project will benefit from and contribute the broader comprehension of the evolution of software in Europe as a technical activity, business, and industry.

In writing the history of early digital computing, scholars have concentrated on the machines, and the electrical engineers who designed their architecture, rather than the programmers who set the machines to calculate problems and process data. In succeeding years and decades, as programming activity took different forms, utilized different methodologies and tools, and facilitated many new applications, the historical concentration has remained on computer hardware. When computer technology—a domain limited for decades to specialists in government, scientific, and educational settings—began to proliferate broadly throughout society with personal computers of the 1980s, programming, in the form of boxed software products, first began to receive widespread attention. In general, software tools, especially programming languages, and software products (boxed software), have been the major topics written about in software history. This has left out many important developments, including a fundamental element of digital computer systems: services. By 2002 the U.S. computing services industry brought in approximately $350 billion in revenue. Today the long dominant computer hardware firm in the world, IBM, derives more than half of its revenue from services, and the proportion of this business to its overall revenue has risen steadily over the past decade. Rarely a day passes without an article or editorial on the threat to lost software services jobs in the U.S. to offshore locations (especially India). Despite this recent and specific attention by journalists, only minimal attention has been given to the long and influential history of computer services. The following project will address this by focusing on the history of computing services from its prehistory, in developing organizational capabilities and customer relationships in office machine services prior to digital computing, through its emergence in the 1950s as a distinct IT business and industry, to its continued rapid growth and prominence over the past half century.

The study will begin with an analysis of how pre-World War II services operations and capabilities of office machine leaders, especially IBM and Burroughs, were translated into their operations in the post-World War II digital computer era. Particular attention will be on the evolution of IBM’s Service Bureau Corporation. The project will explore the role of government, and particularly, the Department of Defense, in funding know-how and resources that were critical to the emergence and growth of the computing services industry. The pioneer computer services firms, including Computer Usage, C-E-I-R, and Computer Sciences Corporation (CSC), soon served a wide range of businesses and government and non-government organizations. The factors behind Computer Usage’s and C-E-I-R’s inconsistency and financial struggles, and CSC’s greater and longer term success will be thoroughly analyzed. With the exception of CSC, other successful emerging giants in services took a far more focused approach in their early years, ADP in accounting and payroll, and EDS in facilities management. The relative strategies of these firms, their development and implementation of technology, the broadening scale and scope of their businesses over time, and overall contributions to the industry will be assessed. Likewise, the advent of a new type of computer service that rapidly challenged the service bureau and batch processing model will be explored: timesharing. This technology built upon and used networking technology and software to create a new paradigm in computing services and resulted in great excitement in the industry and investor community. Timesharing, however, declined rapidly with the proliferation of personal computers (PCs) in the 1980s. The nature of the debate over batch versus timesharing and timesharing versus PCs, and the business and industry ramifications, will be comparatively examined.

In addition to its eventual decimation of timesharing, the personal computer also brought new opportunities and challenges to the services sector as the number of computer users greatly expanded. Whether as a business or a means to sell hardware, IBM, Hewlett Packard, EDS, CSC, and others substantially expanded their services operations overseas over the past two decades. The project will examine these corporations, as well as indigenous services providers in Europe and Asia, from ICL and Capgenini to Wipro Technologies and Fujitsu.

While the focus will be on the large firms, analysis will also address the nature and operations of the small consulting enterprises; the blending of investment and advisory IT services in firms such as IDC, Gartner Group, Forrester and others; and IT consulting giants, such as Andersen Consulting/Accenture, Price Waterhouse, and later, IBM. The study will conclude with an examination of the growing globalization of computer services in the dot.com era, as cheaper and more advanced networking reduced transaction costs and opened up new opportunities for offshoring services. In this period IBM fundamentally transformed itself into a services business. This, at times painful, but ultimately successful transition will be examined within the context of other mainframe firms that made earlier less successful entries into computing services, especially Burroughs and Control Data Corporation. The final part of the study will address the competition, strategies, and execution of major global industry players, including America's Big Six (IBM, H-P, Accenture, ACS, CSC, and EDS); major European players (Capgemini, ICL, and others); and the rapidly growing Indian services corporations (Infosys, Wipro, Tata Consulting, Satyam, and HCL).

The aim of the project will be to provide a thorough, comprehensive history of a major industry that heretofore has been only sparsely examined by historians or other scholars. A focus will be on technological know-how, and the transformation of services as an activity to a business by major hardware corporations to compete with small and mid-sized enterprises that were quick to understand services as an entrepreneurial opportunity. As a result of the deep connections between industry segments, the study will in part revisit and add nuance to the understanding of both the computer and software products trades.

There are rich primary resources for study of this industry, many of which are located at the Charles Babbage Institute (CBI). CBI holds the corporate papers of C-E-I-R, System Development Corporation, Burroughs Corporation, and Control Data. These will be critical to understanding strategy formulation and execution at these firms and how they evolved over time. Much of the discussion and analysis will concentrate on carefully chosen individual firms to gain insights into the broader industry. However, broader quantitative trends in the trade will also be presented and interpreted.

CBI also holds a number of other primary resources fundamental to this project. These include the Market Research Collection, Auerbach and Associate Records, Diebold Group, Inc. Records, and especially, the records of the leading computer services trade association of the 1960s and 1970s: ADAPSO. A number of CBI oral histories also address services related topics, including interviews with John Cullinane, Herbert Robinson, Martin Goetz, Walter Bauer, and others. Archival collections outside CBI will also be important resources, including the IBM Archives, H-P Archives, Sperry Rand Archives, and Apple Computer Archives. Efforts will be made to access the first two corporate held collections, while the latter two are already publicly available at the Hagley Museum and Library and Stanford University respectively. Trade journal literature, especially *Datamation*, *Computerworld*, and *EDP Analyzer*, will also be meaningful sources for the study. Overall, the trade journalists recorded much detail of this important IT industry segment, alongside that of computer hardware and software products. Despite this abundance of high quality material, some areas critical to the services story have yet to be documented, or documentation has not survived. To address areas with less or no documentation, and enhance other areas, two dozen oral histories will be conducted with leading figures in the history of the services field.

Overall, the study is of an industry that has undergone much change, and though important from the mid-1950s forward, has become increasingly central to the overall IT industry. This has occurred for a number of reasons, but two fundamental factors have been the commoditization of hardware and relative attractiveness of services as a business, and the ever growing number and types of computer applications as processing power, networking, and other technologies have rapidly expanded. Both the activity (Burroughs and IBM's pre-digital computing services operations in Europe) and business (nearly all major industry players have European and Asian facilities) of computer services have long had a strong international presence. Over time, the global nature of computer services has grown increasingly deep and profound. Different countries and regions have had far different trajectories. These have been influenced by prior infrastructure, government strategy and investment, military spending, knowledge transfer, access to capital, and a range of other factors. For these reasons a U.S. focused, but fundamentally global, examination of the computer services industry can yield much perspective to the development of software (both as a service and as a product) in Europe.

Literature

Akera, Atsushi, "Voluntarism and the fruits of collaboration: The IBM user's group Share," *Technology and Culture* 42/4 (2001): 710-736.

Alberts, G; A. van den Bogaard; M. Campbell-Kelly; F. Veraart; 'History of the software industry: the challenge', Amsterdam: CWI, 2005 [SEN-R0511, ISSN 1386-369X]

Allais, M. *Erreurs et impasses de la construction Europëenne.* Paris: Juglar, 1992

Agar, Jon. *The Government Machine: A Revolutionary History of the Computer* Cambridge: MIT Press, 2003.

Bauer, F.L., 'Die Algol-Verschwörung', in Hellige, H.D. (Hrsg), *Geschichten der Informatik*, Berlin: Springer, 2004, pp 237-254

Niels Ivar Bech - en epoke i edb-udviklingen i Danmark. Copenhagen: Regnecentralen, 1975 [http://www.datamuseum.dk/site_dk/rc/NIB]

James Beninger: The control revolution, Cambridge/Mass. 1986.

Bemer, R. W., 'A Politico-Social History of Algol', in: Halpern, Mark I. and Shaw, Christopher J., (eds), *Annual review in automatic programming*, volume **5**, London: Pergamon, 1969, pp. 151–237.

Bowden, B.V., ed, *Faster Than Thought. A Symposium On Digital Computing Machine,* London, Pitman 1953

Brillouin, L., Couffignal, L. (1948) *Les Grandes Machines Mathématiques*, Paris, Revue d'Optique.

Campbell-Kelly, Martin. 'The Development of Computer Programming in Britain (1945 to 1955)', *IEEE Annals of the History of Computing* 4(2), 1982, 121-139.

Campbell-Kelly, Martin. *From Airline Reservations to Sonic the Hedgehog: a History of the Software Industry*. Cambridge, Mass: MIT Press, 2003.

Campbell-Kelly, Martin. *ICL: a Business and Technical History*. Oxford: Clarendon, 1989..

Campbell-Kelly, Martin, and William Aspray. 2004. *Computer: a History of the Information Machine*, 2nd edn. Boulder: Westview.

Campbell-Kelly, Martin. 1982. "The Development of Computer Programming in Britain (1945 to 1955)," *IEEE Annals of the History of Computing* 4(2), 121-139.

CECE (1957) *Manuel de programmation pour la machine IRSIA-FNRS*, & *Progress Report N.1*, Brussels, Comité d'Etude et d'Exploitation des Calculateurs Electroniques.

Chambre, P.L., *et al.* (2006) *René J. De Vogelaere. Mathematics: Berkeley*, at http://content.cdlib.org/xtf/view?docId=hb7c6007sj&doc.view=frames&chunk.id=div00013&toc.depth=1&toc.id=

Coopey, Richard, ed. *Information Technology Policy: An International History*. New York: Oxford University Press, 2004.

Cortada, James, *Digital hand. How computers changed the work of American manufacturing, transportation, and retail industries*, Oxford 2003.

Cortada, James, *Digital hand. Volume 2. How computers changed the work of American financial, telecommunications, media, and entertainment industries*, Oxford 2006.

David, R., Duhoux, P., Raucq, J.-P. (1970) *Le Système d'Exploitation OS-360*, Louvain, Université Catholique de Louvain, Centre de Calcul.

Davies, Norman, *Europe.* London: Random House, 1996

Deprit, A. (1960) Les Machines à Calculer Digitales en Grande-Bretagne, *Revue des Questions Scientifiques,* 5(21), 325-345.

Deprit, A. (1994) Portrait d'un calculateur, *Ciel et Terre,* 110, 122-123.

Dijkstra, E.W., *A primer of ALGOL 60 programming*. London: Academic Press, 1962

Edgerton, David, *Science, Technology and the British Industrial 'Decline,' 1870-1970*. Cambridge: CUP, 1996
Edgerton, David, *Warfare State: Britain, 1920-1970.* Cambridge: CUP, 2006.
Edwards, P.N., *The Closed World: Computers and the Politics of Discourse in Cold War America*. Cambirdge: MIT Press, 1996
Ende, J. van den, Kemp, R., Technological transformations in history: How the computer regime grew out of existing computing regimes, *Research Policy* 28(8) 833-851, 1999
ETH (2005) Algol, in *ETHistory 1855-2005* at http://www.ethistory.ethz.ch/rueckblicke/departemente/dinfk/forschung/weitere_seiten/algol/index_EN/popupfriendly
Florizoone, R. (2001) *40 jaar Rekencentrum-LUDIT Geschiedenis*, Leuven, Katholieke Universiteit te Leuven, at www.ludit.kuleuven.ac.be/40jaar/pdf/40jaar.pdf
Folta, Jaroslav, ed., *Computing Technology Past and Future*. Acta historiae rerum naturalium necnon technicarum – Prague studies in the History of Science and Technology, New series, Vol. 5. Prague: National Technical Museum in Prague, 2001
Garr, Nicolas C., *Does IT matter?*, Boston 2003.
Gerovitch, S. ''Mathematical Machines' of the Cold War: Soviet Computing, American Cybernetics and Ideological Disputes in the Early 1950s', *Social Studies of Science 31:2*: 253-287.
Haas, E.B., *The uniting of Europe. Political, social and economic forces 1950-1957.* Notre Dame 2004, 1st print in 1958.
Haigh, T, 'Software in the 1960s as Concept, Service, and Product', *IEEE Annals of the History of Computing* 24:1 (January-March 2002): 5-13.
Haigh, T, 'The Chromium Plated Tabulator: Institutionalizing an Electronic Revolution, 1954-1958', *IEEE Annals of the History of Computing 23:4* (October-December 2001): 75-104.
Haigh, T, 'Inventing Information Systems: The Systems Men and the Computer', *Business History Review* 75:1 (Spring 2001): 15-61.
Haigh, T, 'Research Interactions Between University and Industry in Computer Science in the United States and United Kingdom', 1995 technical report UMCS-95-8-1, *University of Manchester* (UK) Department of Computer Science.
Halleux, R. (Ed.)(2001) *Histoire des sciences en Belgique, 1815-2000*, Vol.2: *De la science nationale aux réseaux planétaires*, Brussels, Renaissance du Livre.
Hashagen, Ulf, Reinhard Keil-Slawik, Arthur L. Norberg, eds. *History of Computing: Software Issues* Springer, 2002
Hendry, John, *Innovating for Failure: Government Policy and the Early British Computer Industry*. Cambridge, Mass: MIT Press, 1989.
Hughes, T.P., Hughes, A.C. eds. *Systems, Experts, and Computers: The Systems Approach in Management and Engineering, World War II and After*. Cambridge: MIT Press, 2000.
Jiřina, Marcel and Korvasová, Květa: "Výpočetní technika" ("Computing Technology"). Chapter 6 in: Folta, Jaroslav (ed.): *Studie o technice v českých zemích 1945-1992 (Studies on Technology in the Czech Lands 1945-1992)*, Encyklopedický dům, Prague 2003, vol. 1, pp. 569-585.
Kaplan, Karel, *Československo v RVHP [Czechoslovakia in Comecon] 1949-1956*. Praha: 1995.
Kruseman Aretz F.E.J., 'The Dijkstra-Zonneveld ALGOL 60 compiler for the Electrologica X1' (*historical note SEN, 2*). Amsterdam: CWI, 2003 [SEN-N0301, ISSN 1386-369X]
Lambert, D. (2000) *Un Atome d'Univers*, Brussels, Racine.
Latour, Bruno, and Steve Woolgar, *Laboratory Life*. London: Sage, 1979.
Magis, T. (2002) *LE fabricant de composants Belge: M.B.L.E.* at http://users.skynet.be/radiocollection/
Mahoney, Michael S., 'The History of Computing in the History of Technology', *Annals of the History of Computing* 10 (1988), 113-125
Mahoney, Michael S., 'The Roots of Software Engineering', *CWI Quarterly*, 3,4(1990), 325-34
Mahoney, Michael S., 'Software as Science - Science as Software', in: Hashagen, U., Reinhard Keil-Slawik, and Arthur Norberg (eds.) *History of Computing: Software Issues* Berlin: Springer, 2002)
Misa, T., Schot, J.W., 'Introduction –Inventing Europe: Technology and the hidden integration of Europe', *History and Technology* **21**-1 (2005), 1-20

Misa, T.J., Philip Brey, and Andrew Feenberg, *Modernity and technology*, Cambridge (Mass): MIT Press, 2003
Mols, S. (1999) *Georges Lemaître. Du Big Bang aux Machines à Calculer*, Louvain-la-Neuve, Université Catholique de Louvain, Unpublished undergraduate dissertation.
Mounier-Kuhn, Pierre, 'L'enseignement supérieur, la recherche mathématique et la construction de calculateurs en France (1920-1970)', in *Des ingénieurs pour la Lorraine - Enseignements industriels et formations technico-scientifiques supérieures, XIXe-XXe siècles* (F. Birck & A. Grelon, dir.), Editions Serpenoise, 1998
Mounier-Kuhn, Pierre-E. "Calculateurs électroniques et nouveaux systèmes d'armes: Interactions Armées/Recherche/Industrie (1946-1959)," in *Colloque: La IVe République et les problèmes d'armement*, ed. Maurice Vaïsse, Centre d'études d'histoire de la Défense–ADDIM, 1998, 376-405.
Mounier-Kuhn, Pierre, "Le Traitement à Façon : Un Survol Historique", *Entreprises & Histoire*, n° **40** (Nov. 2005), p. 52-86.
Naur, Peter, 'The European side of the last phase of the development of ALGOL 60', in: Wexelblat, R.L. (ed) *HOPL-1: The first ACM SIGPLAN conference on History of programming languages*, New York: ACM Press, 1978, pp. 15–44.
Naur, Peter, 'The European Side of the Last Phase of the Development of ALGOL 60'. In *Richard L. Wexelblat, ed. History of Programming Languages.* New York: Academic Press, 1981: 92-118.
Nerheim, G. and Nordvik, H. W. *Ikke bara maskiner. Historien om IBM i Norge 1935-1985*, (Not Just Machines. History of IBM in Norway 1935-1985, in Norwegian). Oslo: Universitetsforlaget 1986.
OECD, *Gaps in Technology: Electronic Computers*. Paris, OECD, 1969.
OECD (1972-1973) *The research system. Comparative survey of the organisation and financing of fundamental research,* Paris, OECD, 2 vol..
OECD, *Software: An Emerging Industry*. Paris, OECD, 1985.
Paju, Petri. "A Failure Revisited: The First Finnish Computer Construction Project. The Establishing of a National Computing Center in Finland." In Bubenko Jr., Janis & Impagliazzo, John & Arne Solvberg (eds.): *History of Nordic Computing.* IFIP WG9.7 First Working Conference on the History of Nordic Computing (HiNC1), June 16-18, 2003, Trondheim, Norway, pp. 79-94. Springer: New York 2005.
Perlis, Alan J., 'The American side of the development of Algol', in: Wexelblat, R.L. (ed) *HOPL-1: The first ACM SIGPLAN conference on History of programming languages* , New York: ACM Press, 1978, 3–14.
Hartmut Petzold, *Moderne Rechenkünstler*, München: Beck, 1992
Pugh, Emerson. *Building IBM: Shaping an Industry and Its Technology*. Cambridge: MIT Press, 1995.
Randell, B and L.J. Russell with a Foreword by E.W. Dijkstra, *ALGOL 60 implementation. The translation of ALGOL 60 programs on a computer*, London: Academic Press, 1964
Salmi, Hannu and Paju, Petri et al. *Written in the Cache. Contributions to the cultural history of Finnish computerisation*, in Finnish. k&h: Turku 2006.
Smith 1994) Bruce L.R. Smith, 'The United States: The formation and breakdown of the postwar government-science compact', in: *Scientists and the state: domestic strcutures and the international context*. Etel Solingen (ed) Ann Arbor: University of Michigan Press, 1994
Termin, Peter (ed.), *Inside the business enterprise*, Chicago 1991.
Thompson, Paul, *The Voice of the Past: Oral History*, 3rd ed. Oxford: OUP, 2000.
Vlček, Jaroslav, *Výpočetní technika v zemích RVHP: ČSSR [Computing technology in Comecon countries: Czechoslovak socialist republic]*. Praha: SNTL., 1975.
Vopěnka, Petr: *Klady a zápory izolované vědy (Positive and negative aspects of isolated science).* In: Markéta Devátá, ed.: *Věda v Československu v období normalizace (1970-1975) (Science in Czechoslovakia during normalisation (1970-1975))*. Výzkumné centrum pro dějiny vědy, Praha 2002, pp. 25-33.
Wijngaarden, A. van, Mailloux, B.J., *et al., Revised Report on the Algorithmic Language Algol 68,* Bern, Springer, 1976
Woitrin, M., *Louvain-la-Neuve et Louvain-en-Woluwé: Le Grand Dessein*, Paris, Duculot, 1987.
Wyatt, Sally, "Non-users also matter" in Nelly Oudshoorn and Trevor Pinch, eds, *How Users Matter*. Cambridge, Mass: MIT Press, 2003.
Yates, JoAnne, *Communication through control*, Baltimore1993.
Yates, JoAnne, *Structuring the information age,* Baltimore 2005.

Yost, S., ‘International Business Machines.’ Encyclopedia of Business History New York Facts on File, 2004

Young, Stephen J, *Real Time Languages: Design and Development*. Chichester:Elis Horwood Ltd,1982.

Zemanek, Heinz: “My relations to Antonín Svoboda”. In: Folta, Jaroslav (ed.): *Computing Technology Past and Future*. Acta historiae rerum naturalium necnon technicarum – Prague studies in the History of Science and Technology, New series, Vol. 5, National Technical Museum in Prague, 2001, pp. 11-23.