The Evolution of ICT by mutual reinforcing loops: innovation, achievements and new paradigms

Chroust Gerhard, Prof. em., Dr.
Donaustrasse 101, A-2344 Maria Enzersdorf, Austria
Kepler University Linz, Austria
gerhard.chroust@jku.at

Abstract: *The Information and Communication Technologies (ICT) have changed hardware, software, industrial applications, business, and gradually our society in their totality. The speed of these changes is due to several reinforcing loops mutually triggering innovations in the respective domains. At the beginning these loops existed between hardware and software, later the other domains also participated.*

Today we observe strong reinforcing feedback loops between the technological capabilities of computers (in all their forms) and human activities in business and society (e.g. e-business, e-learning, knowledge management, social ll computing), leading into the so-called information society.

In this paper we will discuss the major feedback loops between the above domains, identifying innovations in each domain and their impact on one another.

Résumé: *Les Technologies de Communication et d'Informations ont changé le matériel, le logiciel, les applications industrielles, les affaires et progressivement notre société dans sa totalité. La vitesse de ces changements est en raison de plusieurs forts boucles d'asservissement déclenchant mutuellement des innovations dans ces régions. Au debut ces boucles ont existé entre le matériel et le logiciel, plus tard les autres régions ont aussi contribué.*

Aujourd'hui nous observons des forts boucles d'asservissement renforçants entre les capacités technologiques d'ordinateurs (dans toutes leurs formes) et les activités humaines dans les affaires et la société (par ex. les e-affaires, l'e-apprentissage, l'administration de connaissance, l'informatique sociale), ils fait de conduire dans la soi-disant société d'information.

Dans ce papier nous discuterons les boucles d'asservissement importants entre susdit les régions, en identifiant l'innovation dans chaque région et son impact sur les autre régions.

Keywords: *cascading innovations, mutual reinforcing feedback, innovation triggers, social changes*
mots clé: *innovations en cascade, feed-back réciproque, gâchettes d'innovation, changements sociaux*

1 ICT - 60 Years of Evolution

Information and Communication Technology (ICT) has within the last 60 years revolutionalized private and public life as well as society as a whole with speed and profoundness unthinkable even 30 years ago. If we consider the amazing story of success we notice that it is based on a chain of successive key innovations which in turn triggered further innovations in several domains: Hardware Development, Software Development, Globalization and Technical Standardisation, Business Applications, The Individual, Society (fig. 1).

Innovations in one of these domains have impact on the other domains, triggering innovations there, at times *pushing* the use of the innovation into another domain, sometimes *pulling* innovations in a reinforcing way from another domain in order to come to term with urgent needs. We will sketch the interrelation in these domains now supported by or even dominated by ICT. We list innovations which we believe were instrumental in promoting ICT and have an impact and/or a triggering effect on other innovations. According to its nature this listing is very subjective and incomplete. Typically some high impact innovations have only been recognized years after their invention and publication. On occasions the necessary supporting technology was not mature, as in the case of Hypertext, or no need was seen, as in the case of early personal computers. For each innovation we give a short description, followed by an indication of its current impacts (in italic) and its impacts on other innovations (*"Major trigger for innovations described in section ..."*). This paper is a slightly improved version of a presentation at Systemica 2011 [13]. It is based on [12] where more details can be found.

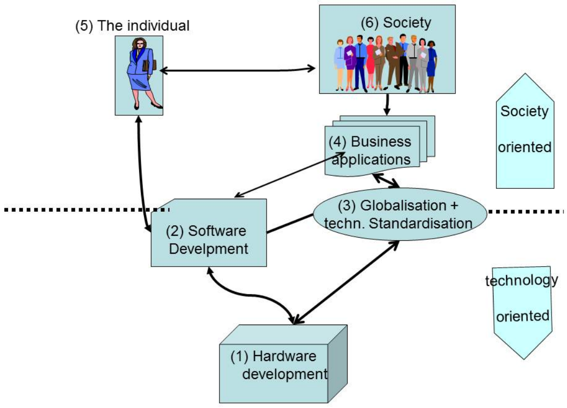

Fig. 1: Feedback loops of impact between different domains of ICT

2 Hardware Development

The basic innovation was the idea of *storing* the intended sequence of steps needed to perform a complex calculation ("the program") in a control unit of the computer, thus making it possible to execute it over and over againat the same time making this program replaceable. The concept of the *stored program* [7] was born. Hardware developers have been busy improving the technology since.

2.1 Innovation: Stored Program Computers

As a result of making the stored program replaceable, the same hardware could be made to behave as a completely different machine. It was soon realized that the computer was able to read these instructions from an internal or external medium, the contents of which could be changed, causing the computer to behave as a *different machine* for each stored program.

This provides individualization of the hardware; the same hardware can be used for widely different applications.

Major trigger for innovation(s) described in sections 2.2, 2.4.

2.2 Innovation: Semi-conductor Technology

Based on semi-conductor materials (initially Germanium, later silicon, and a variety of other materials) both control logic and mass storage can be built in the same technology and can be miniaturized to an unbelievable level. A 'semiconductor' essentially conducts electronic current in one direction and blocks it in the other direction. This property can be used to build amplifiers (transistors) and basic storage elements (flip-flops). The real advantage arises from the possibility of creating large arrays of individual transistors on a single base substrate of silicon ('integrated circuits', 'computers on the chip'). It allows large scale integration (VLSI) with millions of active elements on a small piece of silicon

Mass storage, high-speed logic, and super-miniaturization at dramatically reduced cost make computers small, cheap, and ubiquitous. They replace other technologies and create new applications [26].

Major trigger for innovation(s) described in sections 2.3, 2.4, 3.1.

2.3 Innovation: Automatic Chip Design

The complex programs necessary to design the layout of high-performance computer chips could only be written by using high-level programming languages which had to rely on high-speed computers.

Faster hardware allows larger programs to be executed in reasonable time. This allows running more *complex programs for arranging the elements on a computer chip (the basis of the hardware) which makes chips even more powerful and faster.*

Major trigger for innovation(s) described in sections 2.2, 2.4, 3.2.

2.4 Innovation: Firmware

Around 1960 two dramatically new concepts appeared for the design of computers: Machine Architecture and Microprogramming [2, 8, 23]. The hardware interface is hidden from the normal users by creating a *"machine architecture"* on which the software runs [2]. This is achieved by a set of microprograms (called *"firmware"*) which customize the hardware to present to the user the desired machine architecture. It so-to-speak implements a 'soft hardware interface'. Due to the firmware the user 'sees' usually only the machine architecture.

The computer companies are able to create the architecture independent of the actual hardware. It provides an extra degree of freedom for the computer architects and lets the user switch between different hardware without changes to the software (Fig. 2). All software executes on the machine architecture (Fig. 2).

Software architectures can be designed to optimally fit the application domain, can be adapted over time to changing requirements and can reduce migration problems between different hardware by soft customization of the machine architecture. They can also guarantee persistence of the interface despite of changing hardware.

Major trigger for innovation(s) described in sections 2.2, 3.3, 4.2, 4.3 .

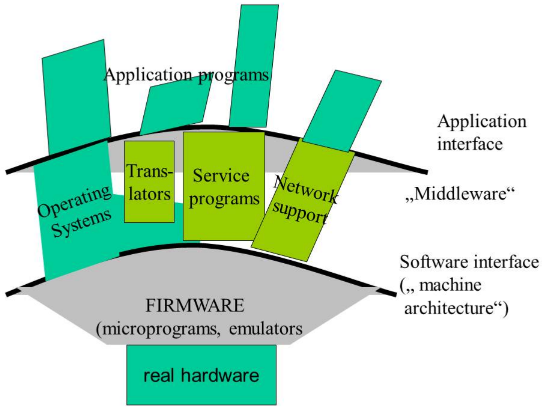

Fig. 2: The layers of a computer system

3 Software Development

Initially programming was a very hard and error-prone job, since every small action of the hardware had to be explicitly specified, e.g. defining from which storage location data should be moved into which computer register. Based on the principle of Unstratified Control [17] compilers were developed which translated higher-level languages (which are more user-friendly) into the machine code (cf. fig. 3). As a consequence software systems were designed in layers (Fig. 2) with the so-called "middleware" providing basic, commonly needed functionality. Software development methods themselves changed, improving quality and productivity.

3.1 Innovation: Unstratified Control and Compilers

A basic idea was Unstratified Control, a term coined by Saul Gorn [17]. Unstratified Control is a typical property of natural languages, it allows us to describe a language (typically its grammar) *by the same language*. The same principle can be applied to programming: A program to be can initially be treated as data and operated on (write, edit, modify, translate, ...) and then used as a set of instructions to control the computer. In stratified control a rigorous hierarchical distinction between data and operations would exists - unstratified control blurs this distinction.

The drudgeries and idiosyncracies of the basic machine language can be handed over to other programs. Domain-specific languages bring programming nearer to the problem domain, pre-fabricated modules can be included in the program etc.

Major trigger for innovation(s) described in sections 2.4, 3.2, 3.3.

3.2 Innovation: Compiler and Higher-level Languages

With the concept of unstratified control a computer program can be *generated* from some other form of expression (e.g. a mathematical description of the algorithm). This allows a desired solutions to be expressed in a so-called problem-oriented language which is more easily understood by humans, especially domain-experts,and allows a more productive solution of problems.

These descriptions are then translated by a *compiler* into the language of the machine architecture, which is 'understood' by the computer [18, 25, 33]. The concept of a compiler was born (Fig. 3). During this translation other administrative tasks (like assigning variables to storage places etc.) are also taken over by the machine. This achieves higher productivity for the programmers and thus the creation of more and more complex programs. The first compiler (FORTRAN) was built around 1974.

Problems can be described and solved in terms of the problem domain. The use of domain-specific languages will grow in the future.

Major trigger for innovation(s) described in sections 2.4, 3.3, 3.5, 5.

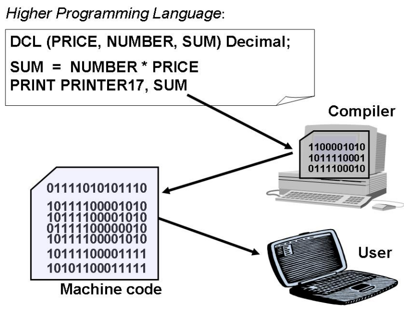

Fig. 3: Use of a Compiler

3.3 Innovation: Middleware/Operating System

A computer has to perform numerous administrative steps from the initial loading and starting of a program, allocating space to the program, performing input-output function, controlling peripheral devices (printers, ...), and cleaning up storage at the end. Soon many of these administrative tasks were handed over to the computer itself: the first operating system [6] was born, typically called 'monitor'. More and more tasks were delegated to this new hierarchical level (networking, administration of subroutine libraries, provision of standard routines, etc.), see Fig. 2. All these functions were collectively called 'Middleware' [27].

Today's sophisticated service-oriented operating systems perform a multitude of common services needed for many software programs, like Microsoft's WINDOWS Operating System.

Major trigger for innovation(s) described in sections 2.4, 4, 5.4, 5.5 .

3.4 Innovation: Standardized Infrastructure

Gradually large portions of tasks are 'pushed down' into the Middleware. These tasks are executed on behalf of all computer programs running on that machine. This relieves programmers from providing these services, reduces the size and the complexity of the application programs and opens programming to a wider population.

Due to the communalization of services they can be build with increased quality, efficiency and usability.

Major trigger for innovation(s) described in sections 2.4, 4, 5.4, 5.6, 5.8.

3.5 Innovation: Software Production Methods

Severe problems concerning productivity and quality in the early 70's have made it evident that software cannot be produced in a haphazard, uncontrolled way. Especially new application and business models have called for speedy, timely, and high-quality systems to be able to meet a small window of opportunity. A breakdown of the central computer and/or data base causes heavy financial losses, even bankruptcy of a company. This threat has had considerable consequences on the software industry which has started to apply methods from other industries to the software development process [22, 34]. It has led to new methods and standards for producing software like component based development [1, 4], Product Line concepts [21], Agile Processes [5], and standards like ISO/IEC 15504.

The new systems approaches promise qualitatively better software, delivered in shorter time at less cost, bringing software production nearer to the standard of other industries.

Major trigger for innovation(s) described in sections 3.2, 5.3, 5.5, 5.6 .

4 Globalization and Technical Standardisation

The growth in communication, locally supported by computers, has enabled technical interconnection, and soon information interchange on a semantic level between different computers. In the meantime globalization and connectivity seem to be the overwhelming effects of computer use.

4.1 Innovation: Digital Convergence

An essential contribution to the success of ICT is that all forms of information, be it text, pictures, music, etc. can be represented in digital form (essentially strings of '0' and '1'). They can be transported over large distances, stored more easily and uniformly, and can also be processed and manipulated by the same (software) tools.

In contrast to analog data one can restore digital data to their original (uncorrupted) values without any loss. Additionally digitalization has proved to be a source for new types of artistic expressions as the yearly Ars Electronica Festival in Linz shows.

Major trigger for innovation(s) described in sections 4.2, 4.3, 4.4, 5.4, 6.3, 6.4, 6.5.

4.2 Innovation: Networking - e-Mail

Merging computer technology with telecommunication technology has allowed for a fast exchange of data, with the DARPA-network [14, 28] being the first. ICT has brought a new form of communication. E-mail is fast, reliable, and asynchronous. It can further be processed seamlessly. E-mail has probably become the most successful networking application. It combines a standard, well-known functionality (letter mail) with an amazing set of additional useful functions. This provides low cost, instant connections, fast electronic transmission of data, recordability and auditability of information.

Despite all its advantages it is also the cause of deteriorating writing and communication style, of an unreasonable distribution of copies, of spamming, and of phishing.

Major trigger for innovation(s) described in sections 4.1, 5, 6.4, 6.5, 7,.

4.3 Innovation: World Wide Web - Information Accessibility

Global accessibility of the World Wide Web has provided new ways of data and knowledge acquisition and dissemination. The possibility for everybody to post at negligible costs arbitrary pieces of information in the public domain offers new possibilities for collecting and disseminating knowledge (and also disinformation!) and has gained an unexpected acceptability from both readers and writers [9]. Information is provided and accessed world wide practically instantaneously at almost zero cost.

Global accessibility of practically all available information is possible As a consequences new ways of data and knowledge acquisition and dissemination have been opened, providing interesting new methods of recherche and cooperation, offering more flexible means of publication.

Major trigger for innovation(s) described in sections 6.4, 6, 4.4.

4.4 Innovation: World-wide Cooperation

Networking also allows computers to be connected over vast distances to solve 'oversized problems' in a distributed fashion. It allows use of slack time in other time zones and the connection of individual PCs into a cooperative work (GRID-computing) to run huge programs (e.g. computation of star movements) in reasonable time.

New types of problems can be tackled by cooperation of scientists in difficult technical or medical situations, by timely exchange of materials (often huge amounts of picture material). A new quality of cooperation evolves. Loss of central control, reduced reliability and higher vulnerability due to external connections cause problems.

Major trigger for innovation(s) described in sections 4.3, 6 .

5 Business Applications

The creativity and ingenuity of the software engineers very soon provided applications which went far beyond the improvement of existing business processes and ventured into domains which previously were the sole domain of the human brain: these types of applications were collectively addressed by the notion of *Artificial Intelligence*.

5.1 Innovation: Quantitative Improvements

Initially computers were simply used to *replace* humans in diverse clerical tasks (book keeping, inventory, registration, etc.). These tasks could be executed much faster, cheaper and with more precision, irrespective of time and day.

This increased productivity and profit allows optimizations, not possible before, e.g. ware house optimization, complex rentability calculations, etc.

Major trigger for innovation(s) described in sections 4, 5.

5.2 Innovation: Process View in Software and Business

The software concept of the execution of programs (i.e. enacting a description of a process) has been carried over to business [31]. Business, too, is seen more and more as consisting of processes, each of them crossing several departments or functional units of an organisation. The focus of interest is the optimal performance of the end-to-end process, not the suboptimisation in one department.

Departement boundaries have become irrelevant, suboptimisation (hopefully) has been reduced. The business processes are more and more initiated, guided, and controlled by computers.

Major trigger for innovation(s) described in sections 5.4, 5.7, 6, 7.3.

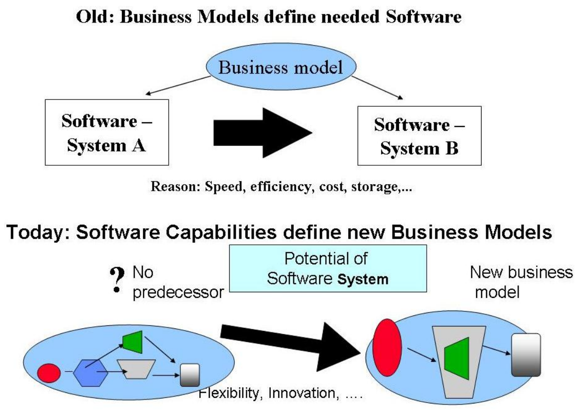

Fig. 4: Old and new relationship between business and software

5.3 Innovation: Software as a Business Innovator

The first computers [7] invented during and right after World War II (Zuse Z1, ENIAC, ...) provided a new tool which was mostly appreciated for its ability to perform relentlessly iterative clericals tasks like accounting, bookkeeping, census, inventory, etc. much faster and with less errors than the human counterpart.

Gradually, however, the applications themselves changed and as a consequence the business world changed. Business models which a few years earlier had been unthinkable, could now be realized in reasonable time and with reasonable cost (Fig. 4), e.g. an electronic bookstore like Amazon.

Today's ICT makes new business models possible which allow for creative use of the new software possibilities. Complicated aggregation, strategies and decision making become possible, including almost instant world wide searches, enabling completely new business models to respond in real-time. The concept of just-in-time delivery, too, heavily relies on several innovations of information technology, like electronic bookstores, micropayments, Google, Recommender Systems, etc.

Major trigger for innovation(s) described in sections 5, 6 .

5.4 Innovation: Web-Services

A global, just-in-time market has opened, providing everything which can be provided electronically to customers. The sellers are located world wide. Competition, comparison of offers, and user information have been improved.

The new forms of services like web services sold/rented via the internet, e-payment are growing fast. Local business will have difficulties to compete.

Major trigger for innovation(s) described in sections 2, 3, 5, 6.

5.5 Innovation: Software Component Market

Initially bespoke software was individually produced, 'hand crafted', and usually not apt to be reused. The rising cost of conventional software development together with the low quality, however, make reuse a necessary new approach. Technological advances enable the cooperation of individual pieces of software, of packing business know-how into individual components and making them available to a broader user community [1, 4]. COTS, Software Components, and Product Lines provide large units of functionality to be acquired and used by third parties with few restrictions.

The multiple use of these components together with a global market [3, 19, 20] induces reduced cost, higher quality, and service orientation. The identification of requirements has to change considerably in order to accommodate available components. It also forces standardization and uniformity for non-core-business applications. Software becomes a commodity.

Major trigger for innovation(s) described in sections 3, 5.6, 5.7.

5.6 Innovation: Open Source Software

Although essentially a philosophical and marketing view of intellectual property [32], Open Source soon became a business factor and even a political factor, as the strife between the European Union and Microsoft has shown [24, 15]. Open Source Software is given away free of charge, it is in the public domain, everybody can use it and modify it with few restrictions [16].

Open Source Software provides independence from large monopolistic software vendors at the price of more responsibility and involvement of the individual user, requiring a 'emancipated and knowledgeable user'.

Major trigger for innovation(s) described in sections 3, 5, 6, 7 .

5.7 Innovation: Electronic Markets

Internet provides access to globally offered goods, especially for goods in electronic form (music, films, downloadable software). The sale of physical goods (cf. Amazon book stores) also profits from the improved business methods and automated logistics, electronic speed, reliability, traceability and auditability. Spamming is unfortunately just an unfair, reckless side effect of electronic markets (see 5.8). Even if it is finally necessary for the end-user to revert to the standard representation, e.g. printing an electronic book at his/her own cost, with e-books circumventing that problem.

Due to its growing number of users, Internet has also increasingly become a prime platform for advertising, be it passive by attractive portals, be it actively by sending messages to potential users and becoming, in its worst form SPAM-mail. Users can access sellers everywhere and often anytime. Shopping is disassociated from physical movement, the market is global. Completely new and innovative products, especially digitized products are offered. Sales efforts are supported by intelligent applications and electronic agents for searching, etc.

Major trigger for innovation(s) described in sections 3, 5.5, 5.8.

5.8 Innovation: Cyber Crime

The speed of computers, the ubiquity of computers and their interconnection, the anonymity, introduce a higher risk, due to ample opportunities for cyber criminals, stealing all kinds of electronic information and blackmailing persons and institutions.

We will see a growing tsunami of spam, malware, phishing, etc. even escalating to cyber war type attacks against countries. The cost for security measures simply explodes.

Major trigger for innovation(s) described in sections 3, 5, 6, 7.

6 The Individual

Both the Internet and the PC provide to the individual user the possibility to utilize individual support functions which go far beyond simple mechanical substitutes for manual processes. The human individual is supported in his/her mental processes. These services provide functionality and intelligence not envisioned even 20 years ago. They have a tremendous impact on the lifestyle of everybody, on communication patterns, interpersonal relationships, etc.

6.1 Innovation: User-centric View

The more every day people get into contact with computers, the more the need for user-oriented applications, especially user-oriented interfaces, becomes a need and a unique selling point at the same time.

The requirements are largely dictated by the market - once the user requirements are really understood, implementation is usually not the problem, given today's performance and cost of hardware. Converting user requirements into appropriate software is still a problem. Human factors and Localization receive growing attention [10, 11]. The useability and user friendliness becomes a the key to market success.

Major trigger for innovation(s) described in sections 5.4, 6.2, 7.

6.2 Innovation: User-oriented Interfaces

Initially the input and output functionality of computers was very poor with respect to human needs and ergonomics. The need to interface with persons without computer knowledge and computer interest, which is a consequence of the ubiquity of cheap computing power made it necessary to modify and re-create user interfaces which fulfil the needs and expectations of normal humans.

Nowadays interfaces attract, even lure, users to use the computer by providing ever more attractive special user-interfaces. Input/output devices cater for different needs, including disabilities, and pure entertainment devices. It also includes consideration of cultural differences [10, 11]. User interfaces are adjusted and modelled after the needs of the (different classes of) users. Specialized new input/output devices are offered.

Major trigger for innovation(s) described in sections 3, 6.1, 5.3.

6.3 Innovation: Personalization of Services

The availability and adaptability of software allows individual users to adapt the systems to their very personal needs and likings. The miniaturization together with low prices allow to supply to users many intelligent products which relieve them from work and at the same time allow instant, interactive information access, often relying on software agents [35].

Personal Digital Assistants and software agents will take over many previously human tasks, anywhere, anytime. The organizations of one's personal life will change dramatically.

Major trigger for innovation(s) described in sections 3, 5.3, 6.2.

6.4 Innovation: Ubiquitous connectivity

By now easy and cheap access to computer facilities (especially via wire-less network connections) is available almost everywhere. Communication both private and business will be anytime, anywhere.

A complete new approach to social communication and interaction will arise, also requiring new ways of behavior and etiquette (blogging).

Major trigger for innovation(s) described in sections 2, 3.3, 5.

6.5 Innovation: Personal Self-Presentation

People have always the desire to present themselves and their achievements to the outside world. Up to now both tradition and technological means have often prevented it. With the new Web-facilities everybody is able to communicate whatever he/she likes to the external world. Everybody can construct his/her own home page and present almost unlimited (and uncontrolled!) information on it.

Obviously the danger of giving up one's privacy exists [29] by allowing others to see and (automatically!) correlate and synthesize information ('the transparent customer'!). This will be both a benefit (health data) but also a danger (abuse of such date, blackmailing etc.).

Major trigger for innovation(s) described in sections 2, 3.2, 3.5, 6.3.

7 Society

Especially the changes with respect to business (section 5) and the individual (section 6) will have profound effects on the society in unpredictable ways. Some indication can already be seen, further emergent effects have to be expected.

7.1 Innovation: Knowledge Dissemination

The instant availability of information, facts, and the low cost of publishing personal opinions (e-publishing) will cause tremendous changes in how humans access, use and evaluate information. This means that education and research will on one hand gain tremendous profits from the global availability and on the other hand must come up with new methods and tools in order to teach, to utilize and to critically evaluate these new offers. A long learning phase with respect to education and research is still needed to be able to adapt the methods and the contents to be taught. Considerable changes [9] [30] are needed.

Information both about facts and new results, will be available almost instantly and globally.

Major trigger for innovation(s) described in sections 5.3, 4.

7.2 Innovation: E-learning, Edutainment and Games

Less emphasis will be put on memorizing facts and data, both will be fetched on-line, just-in-time from appropriate sources. Main emphasis will be put on problem solving, effective ways of finding and analyzing data, performing critical evaluations of data, enjoying the flexibility of access, and accepting life-long learning. Closely related to Edutainment is the fascination of computer games. There is a considerable concern amongst sociologists [29] and psychologists that many individuals are in danger of becoming addicted to computer and games, completely loosing the sense of reality ("lost in the internet").

Know-where will be more important than know-what. Individuals can look it up, timely, accurate, instantaneously. The fascination of computer games - unfortunately very often brutal and inhuman - offers an often overwhelming attraction for the users with potentially related negative influences on life, personality and well-being.

Major trigger for innovation(s) described in sections 5.3, 4, 6, 5.8.

7.3 Innovation: Research

Today researcher have immediate access to relevant (digital) libraries, can quickly distribute their own results and can contact their colleagues on the spur of the moment. Research will be globalized, both with respect to cooperation and competition, and allow for a boundary-free information exchange

This will hopefully eliminate some existing duplications. Innovation cycles will be strongly reduced timewise. Negative examples of plagiarism have already prominently shown up. Another negative aspect comes from overstressing quantity over quality of publications.

Major trigger for innovation(s) described in sections 4.3, 4.4, 6.

7.4 Innovation: Social Computing

This is one of the fastest growing domains, providing connectivity for 'everybody' and creating an enormous demand on speed and capacity of computer storage, networks and computational capabilities.

Socially connecting huge amounts of people will provide new ways of communication, cooperation (Crowd Sourcing) and social interaction.

Major trigger for innovation(s) described in sections 2, 3.5, 5, 5.8, 7.

8 Summary of Computer-related Innovations

The innovations in ICT discussed above depend to a large extent on one another and provide reciprocal support and incentive. These technologies also have impacts on all domains of society, initially in a push mode, but then also creating a pull for more advanced technology. A few conspicuous examples are:

- Faster hardware is the basis for faster software used to create more complex hardware-design programs (chip design!) which in turn allow building of faster hardware.
- More complex software programs allow more complex and innovative business models which need faster software.
- New business models for individuals change the social behavior of the individuals, who in turn require new types of business models.

The interaction between these various domains of our life is indicated in Fig. 1. Changes in the technology-oriented part of the structure cause paradigm changes even in the society-oriented domain of the cycle. The Information and Communications Technologies (ICT) provide us with great hope, with amazing possibilities, but a the same time induce, require or even force changes in our lives, be it private or in business. Many of these changes will have profound effects on our society.

We have to recognize, however, that almost all of the innovations discussed also include considerable risks. Some of the dangers can been reduced by legal and/or technical means, others will have to be handled by social conventions; for many no solution is available in the near future.

References

[1] P. Allen. *Realizing e-Business with Components*. Addison-Wesley 2001, 2001.

[2] G.A. Amdahl, G.M.and Blaauw and F.P.Jr. Brooks. Architecture of the IBM System/360. *IBM J. of Research and Dev., vol. 8, No. 2*, pages 21–36, 1964. definition architecture = p. 21.

[3] N. Asokan, P.A. Janson, M. Steiner, and M. Waidner. The state of the art in electronic payment systems. *IEEE Computer vol 30 (1997, no. 9*, pages 28–35, 1997.

[4] F. (ed.) Barbier. *Business Component-Based Software Engineering*. Kluwer Academic Publishers, 2002, 2002.

[5] K. Beck. *Extreme Programming Explained - Embrace Change*. Longman Higher Education 2000, 2000.

[6] P. Brinch-Hansen. The programming language concurrent pascal. *IEEE Trans. Software Eng. vol. 1*, pages 199–207, 1975.

[7] P.E. Ceruzzi. *A History of Modern Computing*. MIT Press 1998, 1998.

[8] G. Chroust. *Mikroprogrammierung und Rechnerentwurf*. Verlag Oldenbourg, (Feb. 1989), 1989.

[9] G. Chroust. E-teaching - panacea or crisis? *World Futures - The Journal of General Evolution, vol. 59 (2003), no. 1*, 2003. ISSN 0260-4027 WOFUDM.

[10] G. Chroust. Software like a courteous butler - issues of localization under cultural diversity. In *ISSS 2007. The 51th Annual meeting and Conference for the System Sciences. Tokyo, Japan, August 5-10, 2007*. Curran Associates, Inc. (Mar 2008), 2007.

[11] G. Chroust. Localization, culture and global communication. In Goran D. Putnik and Maria Manuela Cunha, editors, *Encyclopedia of Networked and Virtual Organizations, vol II*, pages 829–837. Information Science reference, IGI Global, Hershey USA 2008, 2008.

[12] G. Chroust. Mutually reenforcing innovations in ICT, business, and society. In M. Mulej, M. Rebernik, and B. Bradec, editors, *STIQE 2008 - 9th int. Conference on Linking Systems Thinking, Innovation, Quality, Entrepreneuship, and Environment. Maribor 2008*, pages 19–26. University of Maribor and Slovenian Society for System Research, 2008, 2008.

[13] G. Chroust. The evolution of ict by mutual reinforcing loops: innovation, achievements and new paradigms. In A. Piecq, editor, *8eme Congres International de l' Union Europeenne de Systemique, Oct 19-22. Bruxelles*, page 10. Union Europeenne de Systemique, Bruxelles 2011 (CDROM), 2011.

[14] P. Dickson. ARPA network will represent integration on a large scale. *Electronics vol. 30 (1968), pp. 131-134*, 1968.

[15] S. Forge. Open source software: Unterstützung durch die eu-politik. Technical report, IPTS-Report Ausgabe 85 - European Commission, Joint Research Center 2004, http://www.jrc.es/home/report/ german/articles/vol85/ICT3G856.htm, 2004.

[16] GNU Org. The free software definition. http;//www.gnu.org/philosophy/free-sw.html, [Nov. 2005], 2005.

[17] Saul Gorn. Some basic terminology connected with mechanical languages and their processors: a tentative base terminology presented to asa x3.4. *Communications of the ACM*, 4(8):336–339, 1961.

[18] D. Gries. *Compiler Construction for Digital Computers*. John Wiley, New York, 1971, 1971.

[19] H. Hahn and K. Turowski. General existence of component markets. In *Trappl, R.: EMCSR 2002, Proc. European Meeting on Cybernetics and Systems Research, Vienna, April 2002*, pages 105–110, 2002.

[20] M. Hämäläinen, A.B. Whinston, and S. Vishik. Electronic market for learning: Education brokerages on the internet. *Wirtschaftsinformatik vol. 38 (1996), no. 6*, pages 629–631, 1996.

[21] C. Hoyer. *ProLiSA - An approach to the Specification of Product Line Software Architecturs*. PhD thesis, PhD-Thesis, J. Kepler University Linz, 2007, 2007.

[22] W.S. Humphrey. *Managing the Software Process*. Addison-Wesley Reading Mass. 1989, 1989.

[23] S.S. Husson. *Microprogramming - Principles and Practices*. Prentice Hall, Englewood Cl., 1970., 1970.

[24] Int. Inst. of Infonomics (ed.). Free/libre and open source software: Survey and study - floss, deliverable d18: Final report, part 0: Table of contents and executive summary. Technical report, Int. Institute of Infonomics, University of Maastricht, The Netherlands and Berlecon Research GmbH, Berlin, June 2002, 2002.

[25] P.M. II. Lewis, D.J. Rosenkrantz, and R.E Stearns. *Compiler Design Theory*. Addison-Wesley, IBM Programming Series 1976, 1976.

[26] C.W. Loesch. ICT trends, scenarios in microelectronics and their impact. In P. Doucek, G. Chroust, and Oskrdal V., editors, *IDIMT-2011 Interdisciplinarity in Complex Systems, vol 36, Sept 2011*, pages 369–380. Trauner Verlag Linz, 2011, 2011.

[27] H. Osterle, R. Riehm, and P. Vogler, editors. *Middleware - Grundlagen, Produkte und Anwendungsbeispiele für die Integration heterogener Welten*. VIEWEG 1996, 1996.

[28] E. Plasser and I. Haussteiner. *What Made the Internet Work - A review of interrelations, intentions and the success of the ARPA initiative and its implications for a key technology*. Report no. 8 (June 2005), Publication of the Institute for Systems Engineering and Automation, J. Kepler University Linz, Austria, 2005.

[29] N. Postman. *Amusing Ourselves to Death: Public Discourse in the Age of Show Business*. Penguin USA 1985, 1985.

[30] N. Postman. *The End of Education : Redefining the Value of School*. Vintage Books, New York, 1996, Reprint, 1996.

[31] B. Prasad. Concurrent engineering fundamentals - volume i: Integrated product and process organisation. *Prentice Hall, New Jersey*, 1996.

[32] E. S. Raymond. *The Cathedral and the Bazar*. O'Reilly, UK, 2001, 2001.

[33] P. Rechenberg. Compilers. *Morris, D. and Tamm, B. (eds.): Concise Encyclopedia of Software Engineering, Pergamon Press 1993*, pages 59–64, 1993.

[34] W.W. Royce. Managing the development of large software systems. *Proc. IEEE WESCON, Aug. 1970*, pages 1–9, 1970.

[35] E. Yu. Agent orientation as a modelling paradigm. *Wirtschaftsinformatik vol. 43 (2001), no. 2*, pages 123–132, 2001.