IM 12

ACTA POLYTECHNICA SCANDINAVICA

INDUSTRIAL MANAGEMENT AND BUSINESS ADMINISTRATION SERIES No.12

System Usability of Complex Technical Systems

TEPPO TOSSAVAINEN

Helsinki University of Technology
Department of Industrial Engineering and Management
P.O. Box 9555
FIN-02015 HUT
FINLAND

Dissertation for the degree of Doctor of Technology to be presented with due permission for public examination and debate at Spektri in the Luna Auditorium at the Helsinki University of Technology (Espoo, Finland) on the 10th of May, 2002 at 12 o'clock noon.

ESPOO 2002

Tossavainen T., **System Usability of Complex Technical Systems**, Acta Polytechnica Scandinavica, Industrial Management and Business Administration Series No. 12, Espoo 2002, 236 + 17 pp. Published by the Finnish Academies of Technology. ISBN 951-666-597-7, ISSN 1456-9426.

Keywords: Usability, system usability, complex technical systems, cellular telephone networks, wireless networks, technical fitness landscapes, technical coevolution.

Abstract

The system-wide usability of complex technical systems and particularly the complexity approach is not very visible in usability literature. The reason might be that the usability theories were originally created for computer systems, and ever since this user interface oriented approach has been used regardless of the industry. In this research, the system usability of complex technical systems is comprehensively studied in the case of a cellular network (GSM) by utilising complex system theory, biology, control theory and usability.

In the theoretical framework, we study complex technical systems by viewing literature of complexity research and biology, as well as apply the theories to technological evolution. We study how complexity has been defined and how to reduce user-observable complexity. Following this, we introduce the concept of fitness landscapes and the NK model in engineering in order to study technical coevolution in multiple system cases. Fitness landscape is a useful model, as the product development can be considered as constant hill climbing in a dynamic, ever-changing landscape, where the moves are made by selecting different feature combinations for the product from a vast solution space. By examining technical coevolution in dynamic fitness landscapes, we also see how error and complexity catastrophes arise when the complexity of the product exceeds a certain limit.

The empirical research concentrates on cellular telephone networks by studying the system usability of the networks from the operator's point of view. The empirical results are combined with the theoretical aspects to create new insight for system usability of complex hi-tech products. The construction explains how hi-tech companies live in coevolution with each other and external forces, in this way moulding each other's adaptive landscapes. We also give a definition for system usability with the aid of the effort the users must exert when deploying the system. By using this definition, the system usability of different technical systems could be measured and compared. Finally, we give some guidelines for improving system usability when designing complex technical systems. Good understanding of the described phenomena together with system usability expertise improves the chances of a hi-tech company in developing complex technical systems.

"*Everything must be made as simple as possible, but not a bit more*."

Albert Einstein

Acknowledgements

I would like to express my gratitude to professor Eero Eloranta from Helsinki University of Technology for guiding me through the writing process and allocating time from his very busy schedule. I would also like to thank my opponents professor Klaus-Dieter Thoben from University of Bremen and Dr. Heikki Hämmäinen from Nokia for their good comments in the preview process that have certainly improved my thesis. Special thanks to professor Jukka Ranta from Aspocomp Inc., who in addition to being one of the reviewers, also originally introduced the complexity research to me on a certain course in the technical university. This course, and particularly Kauffman's book 'Origins of order', has had a major impact on me and greatly influenced the whole research. The thesis would definitely not be what it is without the complexity point of view.

I would like to convey my gratitude to professor Anssi Saura from Umeå University and professor Petter Portin from Turku University for checking the biological statements in the theoretical framework and giving improvement suggestions. Professor Saura, for example, read the 'Origins of order' just to be able to give better comments. This kind of commitment is truly remarkable. I also thank Dr. Päivi Eriksson from Helsinki School of Economics and Business Administration for her comments regarding qualitative research methods and Dr. Jan Holmström from Helsinki University of Technology for comments on the guidelines part of the thesis. Special thanks to Ms Tarja Pietiläinen from Helsinki School of Economics and Business Administration for her many philosophical discussions concerning research methodology, ontology, epistemology, theoretical/meta-theoretical issues and, well, everything. I am also grateful to the anonymous informants in the operator companies and Hi-Tech Inc. who were involved in the research.

Last, but not least, I dedicate this book to the person, who induced the change in me many years ago. At the time, I considered it quite a rude awakening – today I thank you. Without you this would not have happened.

Espoo 25 of March 2002

Teppo Tossavainen

Contents

List of figures

List of tables

List of abbreviations

3GPP	The 3rd Generation Partnership Project (3GPP) is a collaboration agreement, which brings together a number of telecommunications standards bodies. The original scope of 3GPP was to produce globally applicable Technical Specifications and Technical Reports for a 3rd Generation Mobile System based on evolved GSM core networks and the radio access technologies that they support.
ACM	Association for Computer Machinery
AIC	Algorithmic Information Content
ASCII	American Standard Code for Information Interchange, comprises auxiliary characters, numbers, letters of the Latin alphabet and special characters.
BCF	Base Control Function. Handles common control functions, e.g. frequency hopping, within a BTS. On a sectored BTS site, there is only one BCF, which takes care of all common functions of the BTSs on this site.
BOPP	Basic Operating Principles and Procedures
BSC	Base Station Controller. Network element in the PLMN for controlling one or more base transceiver stations (BTSs) in the call set-up functions, in signalling, in the use of radio channels and in various maintenance tasks.
BSIC	Base Station Identity Code. Block of code, consisting of the PLMN colour code and a base station colour code.
BSS	Base Station Subsystem. System of base transceiver stations and base station controllers, which is viewed by the MSC through a single interface. BSS is defined as being the entity responsible for communicating with mobile stations in a certain area. The radio equipment of a BSS may cover one or more cells.
BTS	Base Transceiver Station. Transceiver equipment used for communicating with mobile stations in a mobile network.
C	Number of connections between traits of different systems in coevolution
C*	Optimal complexity
CI	Cell Identity. Block of code, which identifies a cell within a location area.
CDMA	Code Division Multiple Access. A radio access technique using codes to distinquish the users.
CM	Configuration Management
DNA	Deoxyribonucleic acid
$\overline{E}$	Effective complexity

ESS	Evolutionary stable strategies
ET	Exchange Terminal, unit on the MSC-side of the switching/transmission interface.
ETSI	Europian Telecommunications Standards Institute
FER	Frame Erasure Rate. Number of erased frames (due to errors in them)/number of all frames.
FM	Fault Management
GPRS	General Packet Radio Service. Mobile service, which gives a packet switched access over GSM to external data networks with high peak transfer capacity.
GSM	Global System for Mobile communication
GUI	Graphical User Interface
HCI	Human-Computer Interface
HF	Human Factors
`HO_margin_PBGT`	Parameter defining how many dB better a new base station candidate must be so that a *power budget handover* to that cell would be possible
`HO_period_PBGT`	Parameter defining how often the *power budget handover* condition is checked
HPUX	Hewlett-Packard Unix
HW	Hardware
IEC	International Electrotechnical Commission
IETF	Internet Engineering Task Force. A large open international community of network designers, operators, vendors, and researchers concerned with the evolution of the Internet architecture and the smooth operation of the Internet. It is open to any interested individual.
IN	Intelligent Network
ISO	International Organization for Standardization
K*	Complicatedness corresponding to optimal complexity
KPI	Key Performance Indicator
LAC	Location Area Code. Part of the location area information (LAI).
LAN	Local Area Network
LAPD	Link Access Procedure on the D-channel
MML	Man-Machine Language. Text-based command language with a standardised structure, designed to facilitate direct user control of a system.
MS	Mobile Station (Mobile telephone). Terminal equipment, which uses radio connection and which can be used in motion or at unspecified points.

MSC	Mobile services Switching Centre. Mobile network element, which performs the switching functions in its area of operation and controls the interworking with other networks.
NK model	A model to study complex systems by using two variables: N = number of components, K = number of cross-connections between the components.
NMS	Network Management System. System for controlling and monitoring the resources of a telecommunications network and recording their use and performance, in order to provide telecommunications services.
NMT	Nordic Mobile Telephone. System of mobile telecommunications developed by the PTTs of Norway, Sweden, Denmark and Finland together.
NSS	Network and Switching Subsystem
OMC	Operation and Maintenance Centre. Maintenance centre of the telecommunications network to which the operations and maintenance of network systems can be centralised.
OMU	Operation & Maintenance Unit
OSS	Operation Support Subsystem
PCM	Pulse Code Modulation. Process in which a signal is sampled, each sample is quantised independently of other samples and converted by encoding to a digital signal.
`pcLowerThresholdsLev`	Parameter defining the limit below which the received power level must fall in order to trigger power increase
`pcUpperThresholdsLev`	Parameter defining the limit above which the received power level must rise in order to trigger power decrease
PLMN	Public Land Mobile Network. A mobile network for the specific purpose of providing land mobile communication services to the public.
PSTN	Public Switched Telephone Network. Telephone network based on circuit switching, including the telephones, local lines, local exchanges, and the complete system of trunks and the exchange hierarchy, which makes up the network.
PM	Performance Management
QOS	Quality Of Service. The collective effect of service performances, which determine the degree of satisfaction of a service user.
$\overline{R}$	Random complexity
R&D	Research and Development
RNW	Radio Network
S	Number of systems in coevolution

SACCH	Slow Associated Control Channel.
SIGCHI	Special Interest Group on Computer-Human Interaction
SW	Software
Tetra	Terrestrial Trunked Radio. Mobile communication network, which is meant for a special group of users, e.g. for one or more enterprises or institutions. Typical examples are oil fields, railways, and public safety and security organisations.
TLUI	Top Level User Interface (in NMS)
TRX	Transmitter/Receiver (transceiver). Combination of transmitting and receiving equipment in a common housing. Network component, which can serve full duplex communication on 8 full rate traffic channels.
$\bar{U}$	User-observable complexity
UI	User Interface. Interface via which a user can interact with software and peripheral equipment that perform operator functions.
w	Individual fitness value of one trait (gene)
W	Overall fitness of the whole system

1. INTRODUCTION

The system-wide usability of complex technical systems and particularly the complexity approach is not very visible in usability literature. The reason might be that the usability theories were originally created for computer systems, and ever since this approach has been used regardless of the industry. Because usability has been regarded predominantly as a Human-Computer Interface issue (see ACM SIGCHI 1992[26]), it has lead to the situation where a lot of complex entities in highly specialised and distributed technical systems have fallen out of the usability studies. As the usability design has not reached all parts of complex technical systems, the usability of these systems has not been very good. However, it is good to remember that many of these systems represent relatively young branches of industry and hence the moderate level of usability is understandable. The standards defining usability (e.g. ISO/IEC 9126-1 2001[1] and ISO 9241-11 1998[2]) have quite broad and covering definitions and would allow a more complete way of understanding usability. However, when interpreted through the present usability paradigm, these definitions do not lead to excellent, comprehensive usability in complex technical systems.

This research set out to study complex technical system usability comprehensively in the case of a cellular network (GSM) by building on complex system theory. In the theoretical framework, we study complexity and nature's way of building complex systems, as well as build an analogy between natural and technological (co)evolution. However, the purpose of the research is not to prove that the natural phenomena in the theoretical framework are true in technological world, although some literature references are pointing to that direction. The intent is to study usability from a new angle, particularly in the case of *complex* technical systems, as well as chart system usability of cellular telephone networks and study the possibilities to improve the usability of complex technical systems. The aim in the empirical part of the research is to uncover things that would not come up in a conventional usability study. As the focus is on system wide usability, the basic idea is to find out what it takes to operate the system and what kind of things require most effort. In addition, the complexity of the system and of the operating work is a point of interest. It would also be good to know how well the presented theoretical concepts apply to technological coevolution, but this is left for future studies to prove.

1.1 Practical value of the research

Telecommunications system industry is a major business world-wide. Manufacturing usable networks that can be operated with little effort has a lot of significance, both to the operators and to the suppliers of these systems. As new features are added to the systems and new generations are introduced, the systems get more complex all the time. The volumes of the networks are increasing every time as a new generation is introduced.

The economic impact could be significant, if the manual effort for operating the large and complex systems is markedly reduced. In addition, this 'lean operating' (Adams/Willets 1996[3]) could enable better competition on the operating and service provision markets.

On the system supplier side, the system usability could also have a major effect. If new guidelines for developing complex technical systems can be found, the competitiveness of the systems could substantially improve. These guidelines could also enable a continued complexity increase without catastrophes. Since this research also aims at increasing the understanding of product development of complex technical systems in general, it could be useful for all suppliers of complex technological products. The study was planned keeping these intended benefits in mind.

1.2 The problem: System Usability of cellular networks

The cellular network suppliers have been doing some usability research, but that also concentrates on Human Computer Interaction issues, mainly in the NMS (Network Management System). In a broad sense, the whole network management system can be considered as a user interface to the telecommunications system, since it has been designed to give the network management personnel a view and means for control over the facilities in the network. A comprehensive system approach comprising the whole system, as opposed to reductionist view, is lacking and some parts of the system, for example, the radio network features in the BSC (Base Station Controller) have been completely spared any usability studies.

In this research, we aim to create new contribution for usability, which would be particularly applicable for complex technical systems. Another objective is to approach complex system development with the aid of complexity doctrine, fitness landscapes and coevolution. Thirdly, we aim to find ways to improve system usability of these systems by applying the theoretical framework.

Since it may not be easy for a person without any background in cellular networks to grasp the theme of this research, it is useful to draw an analogy with something more familiar, for example, buying a car. It is quite obvious that this analogy does not do complete justice to cellular networks, because the complexity of these networks is far beyond the complexity of any car. In addition, the cellular networks have not existed as long as cars, which is another reason why the usability of the networks is so modest. There is still a lot to learn. However, bearing these things in mind, a humoristic short story characterising the situation might be very enlightening for many readers. The story is presented in Appendix 1.

1.3 Objectives and research questions

The research approach is hermeneutic and the objective is to create insight for high-tech companies in order to build more usable technical systems. Although, the research has been done in the cellular network domain, some of the results can also be applied to other complex technical systems. However, the purpose of the research is not to prove this.

There are three main research questions in this study, and the questions are explored in three different ways. The first two questions are related to the theoretical framework and studied by theoretical means, the third question having three sub questions is studied both with the aid of a cellular network simulator and empirically. In the following, we shall take a look at these questions.

The main theme of the research is system usability. On the other hand, experience has shown that many technical systems are complex – particularly in the researcher's experience the cellular networks are quite complex systems. Therefore, we set out to study if there is a connection between the two. While doing the theoretical framework exploration by gathering theories of natural complex systems and complexity research, we decided to find an answer to the following questions within the domain of complex technical systems.

Q1 How to understand system usability of a technical system via its complexity?

Q2 How to improve system usability by managing complexity?

These questions are used as a guideline in the theoretical framework creation. A definition for system usability is also created at this stage prior to the empirical research. To validate the theoretical aspects of Q1 and Q2, and also to create practical utility for this research, we also asked:

Q3 What is the status of system usability and complexity in cellular networks and how could the system usability be improved?

- Are cellular radio networks *complex* technical systems?
- What is the present system usability of cellular networks?
- In case it is not good, can it be improved by managing complexity?

The whole research mission consisted of four steps. First, to chart the present situation of usability thinking in order to see whether some new contribution is required, second, to check if a good theoretical framework already exists or whether we should do an exploration in that area, third, to chart the present system usability of cellular networks and fourth, to improve the system usability of these systems. The reason why cellular networks was selected for the subject of the empirical research is quite natural, since the author has worked for a network provider for a long time and thus the area is familiar. The second reason is that it is useful for the employer. The third reason is that the domain is interesting to study from a new angle.

The first question when starting the research was: *What is the present paradigm in usability?* This was studied by reviewing books representing the present usability literature. A more thorough review was done later (see Usability in chapter 2 'Definitions'). Based on the literature review, it was concluded that the usability research so far has concentrated mainly on either computer systems with the emphasis on Human-Computer Interaction (HCI), or small, simple devices such as remote controllers and heart rate monitors, with very little research to cover comprehensive usability of complex, distributed technical systems. There are many different things in addition to human-computer interfaces, which greatly affect usability in complex technical systems. If these things are not considered, the usability cannot become very good.

Hence, the question arose: *What can we do to change the usability orientation towards a more comprehensive approach?* If we go to the field and perform a usability research based on the existing usability paradigm, how much can we improve the usability of complex technical systems? Alternatively, if we create a different kind of theoretical framework, on which we then build the research, is more substantial improvement more likely? We ended up with the latter alternative. Material for the emerging framework was searched for by combining theoretical insight from complexity research, biology, control theory, technology and usability.

We also wanted to know, if cellular radio networks are complex technical systems. This was studied by searching for interdependencies between radio network parameters with a network simulator. Simulator was selected for the task, since the operators would not allow playing around with the live network parameters. It is also easier to show the complexity in this manner than doing time consuming experiments in the real network. The down side of a simulator exploration is that one can never be sure that all the necessary phenomena will emerge in the simulation. For example, if some interdependencies in the real world are caused by the non-simulated phenomena of the external world (on the radio path), those results will not show in the simulation. In addition, we did not have much resources for this kind of experimenting, which is a huge task in the sense that the number of different parameter combinations is astronomical. However, we were lucky enough to find some interdependencies even with this rather limited search.

The next task was to empirically chart the system usability of cellular networks the way operators see it. While doing the empirical study, we also searched for engineering improvements to see how easy it is to improve the current situation. Providing that the empirical result shows modest system usability, finding easy improvements would also support this result. After charting the system usability of the present cellular networks, we search for improvement possibilities by building on the theoretical framework. Consequently, we apply the coevolution and complexity theories to product development of complex technical systems and create a set of guidelines aiming for better system usability.

1.4 Methodology

In this subchapter, we discuss the methodological issues related with the research. From the very beginning a main theme in hermeneutics has been that *the meaning of a part can only be understood if it is related to a whole* (Alvesson/Sköldberg 2000[4]) (see Figure 1). This fits particularly well in complex systems that are considered to possess emergent properties (see Bertalanffy 1984[95]), which cannot be explained by the properties of the parts (see chapter 3 'Complexity'). The basic idea in hermeneutics also concerns *the revelation of something hidden*, rather than the correspondence between subjective thinking and objective reality (Alvesson/Sköldberg 2000[4]). Even though we have to note that this research is mainly one person's view to the situation, it is not of paramount significance. The very act of understanding is primary; subject and object are secondary categories. Understanding is considered to be the central purpose in post positivist research paradigm (see Lincoln/Guba 1985[197]).

Understanding comes more from the act of looking over the shoulders of actors and trying to figure out (both by *observing* and by *conversing*) what the actors think they are up to (Geertz 1979[5]). Understanding thus entails a kind of empathic identification with the actor (Schwandt 2000[6]). In this research, the empathic approach has been used both when doing the research field work (contextual inquiry) and also as a suggested guideline to improve the system usability of complex technical systems (see chapter 8.3.1 'Empathic feature research'). The data acquisition methods were selected based on ethnographic point of view, which was deemed suitable for the task, since it allows the empathic identification. A major point in ethnographically inspired approaches is that work is a socially organised activity, where the actual behaviour differs from how it is described by those who do it. This implies that detailed studies of work must include *observations* as well as *interviews* (Simonsen/Kensing 1997[7], Blomberg et al. 1993[8]).

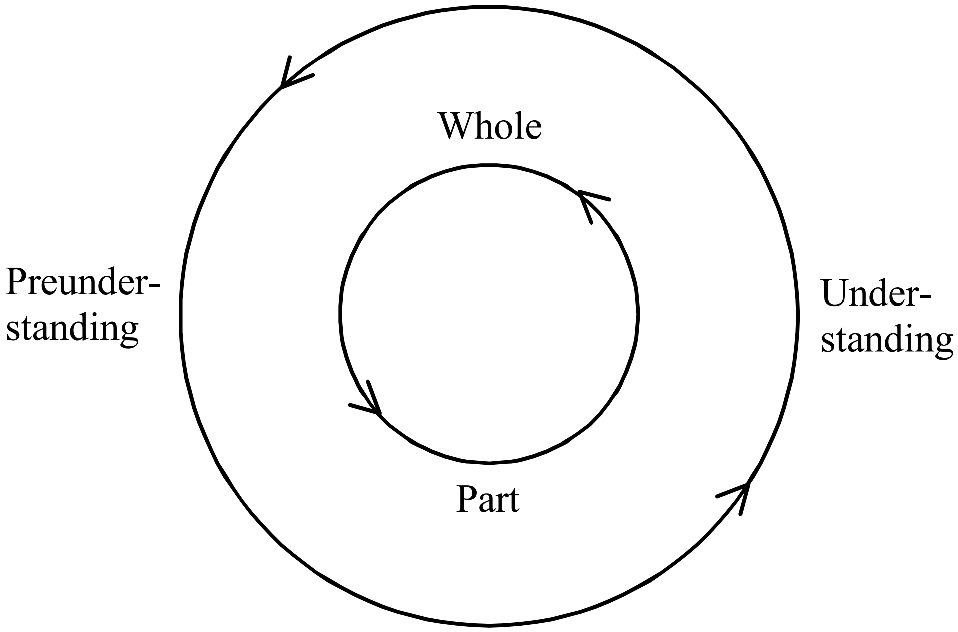


Figure 1. Hermeneutic circles: The part can only be understood from the whole and the whole from its parts. Understanding must continually refer back to earlier pre-understanding and pre-understanding must be fertilised by the new understanding. Source: Alvesson/Sköldberg 2000[4].

Understanding a new thing requires pre-understanding, yet at the same time, pre-understanding – if it is to be developed – requires understanding of the whole (Alvesson/Sköldberg 2000[4]). A prior ethnography (observation for a lengthy period before the study) is recommended for gaining good pre-understanding (see Lincoln/Guba 1985[197]). The author has been working in the field of cellular networks for more than a decade. This experience consists of software design, radio network planning, as well as product and program management. Particularly, the many years spent in cellular network planning doing coverage, capacity, frequency and parameter planning as well as optimisation for different networks, such as GSM, NMT and TETRA[a], has given a profound pre-understanding of the technology and complexity of the systems. As the networks are tailor-made individual items, the planning work is done in the target country in close co-operation with the operator. This also gives a good perspective to the operator's world and problems. The good pre-understanding of the technology, cumulated through the years, was useful throughout the research, particularly in the data collection and analysis phases. This pre-understanding of the technology accelerated the gaining of understanding when doing the actual research.

To understand presupposes pre-understanding, but at the same time pre-understanding is an obstacle to understanding (Alvesson/Sköldberg 2000[4]). The author had this kind of dilemma with the usability pre-understanding. On the one hand, when going to the field it would be good to know the usability theories in order to study the technology from that angle. On the other hand, when doing this, one's mind becomes contaminated by the present paradigm and it is hard to break free from that mindset. Hence, to minimise the influence of the present usability paradigm, the author decided to only use the newly created theoretical framework (in addition to research methodology) when entering the field, and do a proper usability literature study *after* the field research. Of course, the author had some kind of pre-understanding of usability to start with (e.g. based on the quick survey prior to the empirical study), but it was not expanded before the field work. Hence, the pre-understanding/understanding circle was in effect already before starting the empirical research, at least in the form of developing the pre-understanding.

The method selection also raised a similar kind of contemplation. In scientific inquiries, a crucial step is to ask the right questions. Each question contains presuppositions, largely implicit (Bohm 1980[9]). Using a structured approach in the interviews can provide more reliable, quantifiable data than an open-ended interview, but considering the researcher's strong pre-understanding of the technology and systems, using predetermined questions drawn prior to the field work could have biased the questions and perhaps left out things that the informants deem important. If one is interested in an area of study in which the information sought is relatively limited, then there is every reason to think that the use of focus groups or a fixed-choice questionnaires might be appropriate. But if one is interested in questions of greater depth, where the knowledge is often taken for granted and not readily articulated by most members, or where different individuals or groups involved in the same line of activity have complicated, multiple perspectives on some phenomenon, then in-depth interviewing is likely the best approach (Johnson 2002[10]). In addition,

---

[a] Terrestrial Trunked Radio. Mobile communication network, which is meant for a special group of users, e.g. one or more enterprises or institutions. Typical examples are oil fields, railways, and public safety and security organisations.

the intention was to create a new way of understanding system usability, which was considered to need an open-minded approach. Therefore, open-ended (in-depth) interviews were selected for the research.

In this research, the major part of the theoretical knowledge is created first to form the theoretical framework, whereas the problem starts to reveal itself as the empirical evidence builds up. The theoretical framework is also further refined in the process and then used as an aid when interpreting the empirical data. The new element in the study is using *natural systems* and *complexity research* as a starting point to create the approach for looking at complex technical systems. This view is then used to create the results. When developing the practical implications based on the research results, the influence of the theoretical framework is also very strong. The solution tries to capture the essence of complex technical systems, complexity, and handle the issue to the benefit of the system.

The research approach in this work is hermeneutic, as it provides a means to understand the development of complex technical systems and usability by the way we understand *nature* building complex systems. It also aims for understanding usability of complex technical systems with the aid of *complexity theory*. To create such insight, the researcher empathically applies (Niiniluoto 1999[11]) the initially created theoretical framework to the phenomena found in the empirical study. In other words, the theoretical framework is used as a lens through which the empirical data is interpreted. In hermeneutic research, the relationship of the researcher and researched becomes important, as the objective is to increase understanding of the researched phenomena. The research could have been done by any researcher, but without the pre-understanding of the studied technology and also the novel theoretical framework, the results would probably have been very different. For example, the things that the researcher asks in an unstructured interview and particularly the things he considers significant enough to ask additional questions about, are different for people with different backgrounds.

Many times in the natural sciences, the thinking is inclined towards the 'objective truth', which the researchers are looking for. However, in philosophical hermeneutics, understanding is interpretation. Understanding is participative, conversational and dialogic. It is something that is *produced* in the dialogue, not something *reproduced* by an interpreter through an analysis of that, which he seeks to understand. In this sense, philosophical hermeneutics opposes a naïve realism or objectivism with respect to meaning and can be said to endorse the conclusion that there is never a finally correct interpretation (Schwandt 2000[6]). In this research, we produce an interpretation of cellular networks as complex technical systems and study the system usability of these systems based on the created theoretical framework. This interpretation is, by no means, the only 'objective truth', but it provides a new way of approaching complex system usability thus increasing our understanding of these systems. However, natural sciences usually follow the tradition of seeking the ultimate truth. Consequently, we also validate the results with a panel of industrial experts to see how well the interpretations are accepted in the cellular network industry. In other words, if the created interpretations are accepted by the experts of the industry in question, it is likely that an adequate description of the truth has been found.

1.5 Field work and analysis

In this sub chapter, we describe the execution of the empirical research and the data acquisition methods selected for the research. We also study how the data acquisition methods were actually used when doing the field work. There were totally four network operators involved in this research, three of which were European and one Asian. The idea was to select operators with different amounts of experience and network size (see Table 1). The fifth element in the table is a network equipment supplier (the name has been changed) that designs and manufactures cellular telephone networks for the network operators.

Table 1. Operator background and number of interviewed employees.

Operator A	Operator B	Operator C	Operator D	Hi-Tech Inc.	
Medium	Large	Large	Small	N/A	Network size
High	Medium	Low	High	N/A	Experience/skills
15	10	7	5	9	Interviewed people

There was some difficulty in arranging access to the operators for the research. Some of the operators totally refused access. Some of them agreed partially – it was possible to arrange the interviews, but without observations. As a result, we decided to do the observations with one operator, which gave full access to observations in the network control room, and form the cases based on the observed operating tasks of this operator. Then interviews with the other operators were used to complement and verify the information gathered from the cases. Some interviews were also arranged with the first operator, but those interviews are not presented separately in this report – they have been combined with the observed cases and used as a means to increase understanding.

The cases in this research are operating tasks (see chapter 6.8 'Operator A'). The selected eight tasks are among the most common procedures, which every operator has to deal with. Although, the working practices are slightly different with different operators, it is possible to note the difficulties in operating by first studying closely the operating processes of the first operator and then checking with interviews whether the other operators are facing similar difficulties. Gaining good understanding with the operating tasks of the first operator also helps in understanding the problems the other operators tell about.

The research utilised mainly two different data acquisition methods: unstructured interviews and participant observations in the users' work environment. In addition, some operating process flow charts of Operator A and written documents of Hi-Tech Inc., for example, basic operating principles and procedures for Fault Management (FM BOPP 1998[161]), Configuration Management (CM BOPP 1998[162]) and Performance Management (PM BOPP 1998[163]) were used to complement the study. In chapter 6.7, there are short descriptions of the mentioned management areas derived from the documents.

All interviews and observations were recorded either on a minidisk or a micro cassette and some written notes were done for later analysis. The material was then transcribed. The transcriptions do not contain everything word-by-word, but some direct quotations were written down the way they were said to present some samples of the data. The purpose of the quotations is to give the readers the opportunity to determine whether the interpretations made reflect the real situation. The quotations also enable the reader to better catch the situation and thoughts of the system users. The case descriptions based on the data from Operator A were given back to the operator, where two different persons read them and made some corrections. The informants also told about the improvements, which were made after the initial field research. Those are listed in appendix 3.

In the validation phase, a panel of industrial representatives evaluated the practicality of the main contributions. The panel contained managers from Hi-Tech Inc., a manager from a network operator and an independent researcher having experience of other complex technical systems besides cellular networks. A summary of the comments can be seen in chapter 10 'Validation'.

The proposition for the research was to explore complex system theories to find out how usability can be studied with the aid of complexity, find out the present usability status of cellular networks and study the ways to improve it by using the theoretical framework. With these guidelines we formulated the research questions and the design of the study. The research questions and the newly created theoretical framework then guided the analysis of the data.

On the high level of analysis, pattern matching (Yin 1994[12]) was used as a dominant mode. This means that the researcher was looking for certain predicted patterns or outcomes in the data to find if evidence for the prediction can be found. At the same time an alternative possibility was explored. The initial prediction was that a) the usability of these systems is not very good, b) the systems are quite complex, as well as distributed and require a more comprehensive usability approach than has been used. The alternative was that a) the usability of the systems is fine and hence b) the conventional HCI-oriented usability is sufficient for these complex technical systems. However, most things looked for in the research would most likely not be visible at all in a usability research based on the present HCI-oriented paradigm. The searched outcomes were defined on a rather high level a priori to avoid too narrow-minded approach.

Since the author has gathered most of the work experience within the company, whose system was involved in the research, the internal connections provided an access to many things concerning the product development processes. This background knowledge is voiced through some footnotes in the case descriptions and also used in the data analysis.

Reporting of the research follows the Linear-Analytic structure, where the sequence of subtopics consists of problem introduction, methods of exploration, relevant prior literature, findings from the data collected and analysed, as well as the conclusions and implications (Yin 1994[12]).

1.5.1 Interviews

Yin 1994[12] argues for interviews as one of the most important sources of information in case study research. According to him, the case study interviews are most commonly of an open-ended nature, so that the interviewer asks for the facts of a matter, as well as for the respondent's opinions about the events. The interviewer may also ask the respondent to propose his or her own insights into certain occurrences and use the propositions as the basis for further inquiry. ETSI specification ETR095 1993[13] also mentions interviews as one of the recommended research methods in connection with telecommunications systems.

Interviewing is a technique for gathering information about usability issues by talking directly to the users. An interview can typically gather more information than a questionnaire and go into a deeper level of detail. Interviews are good for getting subjective reactions, opinions, and insights into how people reason about issues i.e. qualitative subjective data. (UFG[14]). In interviews, as opposed to surveys and questionnaires, the researcher is present to interact and facilitate discussion about the issues raised by the questions. With multiple users present, as with focus groups, the interaction among the users may raise additional issues, or identify common problems that many people experience (Hom 1996[15]).

Structured interviews are ones with a pre-defined set of questions and responses. They are sometimes better than questionnaires, because thorough response is usually easier and because optional avenues of questioning, based on answers to earlier questions, can be explored. A structured approach can provide more reliable, quantifiable data than an open-ended interview, and can be designed rigorously to avoid biases in the line of questioning (UFG[14]). On the other hand, it can be that the researcher gets answers to the wrong questions, if everything is frozen beforehand.

Open-ended (unstructured) interviews permit the interviewer to ask broad questions without a fixed set of answers, and explore paths of questioning, which may occur to the interviewer spontaneously during the interview. Moreover, they allow the respondent (interviewee) to provide additional information. An open-ended approach allows for an exploratory approach to uncover unexpected information, used especially when the exact issues of interest have not been identified yet (ibid.). In this particular research, where a new approach for the researched topic is sought, this is very important. The researcher must be careful not to repeat the old paradigm.

The primary research method in this study was the unstructured interviews (ETSI specification ETR095 1993[13]), which were used with all studied operators. Representatives of many different working groups of the operator personnel were interviewed. Originally, the idea was to use people with very different lengths of work experience for the interviews, but soon it was discovered that the inexperienced employees do not have much to say. Therefore, for the remainder of the interviews only people with relatively long experience were chosen. The interviewed employees consisted of, for example, network management specialists, BSS system experts, network planners, commissioning engineers, network performance engineers etc. Additionally, some experienced people from Hi-Tech Inc.'s organisation, who were working closely together with the operator, were also interviewed. These employees consisted of, for example BSS technical support engineers, network planners, field managers and a customer service centre manager.

When doing interviews, there is always a danger to fail to get the right questions asked (Stake 1995[16]). In a structured interview, also the pre-drawn set of questions limits the interviewee's story, so that they cannot freely select the essential topics. The true, internal voice of the subject comes through only when it is not externally screened or otherwise communicatively constrained (Gubrium/Holstein 2002[17]). The researcher tried to avoid these problems by deciding only the themes for the interviews in advance letting the informants decide what is relevant. This method of interviewing is called the *general interview guide approach* (Patton 1990[18]). In addition, different informants were not asked to tell exactly about the same things (Stake 1995[16]), as the expertise of the interviewees is limited to a sub-domain close to their own work scope. The interviews were started by asking the users what they normally do while doing their daily operations. When the informants were telling about their normal everyday work and tasks that they are involved with, the interviewer asked additional questions and clarifications about things that were interesting based on the theoretical framework and experience. In addition, the opinions of how the interviewees feel about using the system for their work tasks were probed. Many problematic things came up and the users were also asked to give improvement suggestions for the problems. The interviewees were also asked to tell about their relationship with other working groups in the organisation (see Tashakkori/Teddlie 1998[19]). In this way, it is possible to build a comprehensive picture of the whole operating work.

The interviews covered altogether 37 people from four different operators in Europe and Asia, as well as 9 people from Hi-Tech Inc.'s organisation (refer to Table 1 above). The interview sessions were all face-to-face sessions and lasted ½…2 hours depending on the experience and knowledge of the participants. Almost all sessions were individual interviews. There were two exceptions. With one of the operators, there was one interview featuring two people, because the interviewees were going to a meeting and wanted to save some time. The other exception happened with another operator, where the interviewees were a little uncertain of their language skills and wanted to come together, so that they could support each other.

1.5.2 Observations

Observations of the technology at work are invaluable aids to any further understanding of the limits or problems with the technology (Yin 1994[12]). When doing interviews it is not possible to obtain all the useful information from the informants, because they do not remember all the things they are experiencing in their work. Hence, it is good to use a method, which closely follows the actual work tasks. *Observing* the employees at the work place dealing with the daily routines gives information not coming up in interviews. Observation was selected for the second data acquisition method, and it was done by using contextual inquiry to further enhance the information flow in the observation situation.

Contextual inquiry follows many of the same process steps as field observations or interviews. Different considerations are kept in mind, however, with some portions of the process. For example, interviewing during a contextual inquiry study usually does not include set, broadly worded questions. Instead, the partnership between the interviewer and interviewee is used to create a dialogue, one where the interviewer can not only determine the user's opinions and experiences, but also his or her motivations and context (Hom 1996[15]).

Contextual inquiry is based on three core principles: that understanding the *context* in which a product is used (the work being performed) is essential for elegant design, that the user is a *partner* in the design process and that the usability design process (and assessment) must have a *focus* (UFG[14]). In the observations, the context was understood via the operating tasks. Only by gaining understanding in what is to be done and why, one can form a view of the technology required for the tasks. Anyway, the context was somewhat familiar already when starting the research, as the researcher had been working in close co-operation with the operators before. The partnership manifested itself in the discussions during the observation, when the researcher tried to gain the essential characteristics of the tasks and problems, as well as the users' evaluation of the technology and reasons why things are done the way they are. The focus in the field work was to learn what kind of tasks the operator is performing and what is causing most effort in the work.

When considering a practical approach to the study, certain methods of how to carry out the contextual inquiry should be kept in mind. Observe the users in the real-world environment and maintain a concrete approach. Concentrate on what the user is doing or just did keeping the dialogue in the context of the assessed situation. When observing the workflow, some important issues should be noted, mainly to determine what the common tasks are and how they are done. Also important to notice is why the users perform the tasks the way they do, can they efficiently utilise the system, how many different tools do they have to use, are they using many additional tools, could there be more efficient ways to achieve the goals, is the system somehow preventing more efficient processes, how much manual effort is involved and how much communication does it require between the different groups in order to perform the tasks.

To see how the users actually handled the operational tasks in practice, the observations were performed in the network control room by using the contextual inquiry method. When doing the observation, the operating personnel were handling their normal routine tasks, but the employees were asked to tell what they are doing and why, as they went along. The researcher also asked additional questions, when the employees proceeded step by step. This is an elaborate way of learning about the problems – it is easier to understand the troubles in the real context. It was very illuminating to see all the laborious procedures the people must perform and how much mutual communication s and synchronisation with other work group some of the processes require, since the tasks of different groups are so much interdependent.

This seems to be a very good way to get the user feedback, since the users can easily point out the down sides and shortcomings of the system when using it for the normal work tasks (Rasmussen 1994[20]). The context brings many things to mind that are not thought or mentioned in an interview (Nickerson/Landauer 1997[21]). If the users are only asked to tell their views in an interview, they cannot remember very much of the problems they have encountered and even when they do, they cannot elaborate on the problems, as they can in the real work environment.

These observations were performed with one of the participating operators in two different cities and they lasted seven days altogether (three days + four days). Many times these sessions turned out to be interesting conversations about the tasks and how the system should actually support them. The users also had many suggestions to improve the system to better support them in the work. The observed people were mainly network control room employees, who monitor the operation of the radio network, commission the new base stations in co-operation with the field group and do all kinds of alterations in the BSS[a], for example BSS splitting and base station equipment changes operations.

To increase the reliability of observational evidence, a common procedure is to have more than a single observer making an observation (Yin 1994[12]). In this study, there was another person in addition to the author participating in the observations. The observations were recorded on audio minidisk and written down for later analysis. When analysing the written data, case descriptions were drawn for the most common tasks, which all operators most likely have to perform in order to build and run their networks. The information from the interviews of the same operator was used to build a better overall picture of the operations and complement the cases.

1.6 Overview of the thesis

The thesis starts with definitions of the main concepts including a literature review, which covers the body of knowledge in usability research. The purpose of the review is to chart how the present paradigm understands usability. As the present usability theories are created for computer systems, the author wanted to create an approach particularly suitable for complex technical systems. This System Usability concept is defined after the usability literature review.

In the theoretical framework, we are trying to catch the essence of technological evolution of complex technical systems by applying the theories of natural complex systems and some other

---

[a] Base Station Subsystem, see also 'Cellular networks' in chapter 6.

fields to technological evolution. The empirical research concentrates on cellular telephone networks by studying the system usability of the networks from the operator's point of view. The final insight is then created by combining the theoretical considerations with the empirical results.

In chapter 1, we first introduce the reader to the motif of the research. This also includes a short story, which can be found in appendix 1. Then we explain the objectives of the research and present the research questions. We also discuss methodological issues, as well as describe the used data acquisition and analysis methods in this chapter.

In chapter 2, we define some basic terminology and take a look at the present paradigm of usability in the literature, after which we define the concept called System Usability.

In chapters 3 and 4, we create a theoretical framework for the research. First, we investigate how complexity has been defined, how nature builds complex systems, and study the ways to reduce user-observable complexity. There is also a short description of complex adaptive systems. Following this, we introduce the concept of fitness landscapes and the NK model in order to study complexity and product development of complex technical systems. We also see how error and complexity catastrophes arise, when the system size exceeds a certain limit. Finally, we examine technical coevolution in dynamic fitness landscapes.

In chapter 5, we draw a conclusion of the concepts in the theoretical framework and sum up the means for improving system usability of complex technical systems, as well as study the meaning of the theoretical framework.

In chapter 6, we present the empirical part of the research, starting with a short introduction to cellular networks and product development of cellular networks as it is done in Hi-Tech Inc. Then, we present some simulation results to demonstrate that cellular radio networks are complex technical systems. This is done by showing the interdependency of certain radio network parameters. Next, we have a look at the typical work groups that are required for the operational tasks of cellular networks, as well as review the functions of the network management group and a network management system. Following this, we study the practical operating tasks with the aid of case descriptions and present some preliminary analysis of the data.

In chapter 7, we present the findings from the empirical research gathered from all the operators in the study.

In chapter 8, we combine the insight of the theoretical framework with the empirical data and create a collection of guidelines to improve system usability in industry.

In chapter 9, there is a summary beginning with an assessment of the results in relation to the research questions. Then, we review the main contributions of the research consisting of three parts. First, there is a definition of system usability. Second, we present a compilation of guidelines to improve system usability of complex technical systems. Third, we produce a conceptual model of technical product development in a coevolutionary situation.

In chapter 10, we present author's assessment of the validity and sum up the comments of some industrial representatives concerning the results and suggestions.

In chapter 11, we discuss the results of the research and present some topics for further research.

2. DEFINITIONS

In this chapter, we define some essential central concepts. For example, fitness is used quite extensively in this research to measure the 'goodness' of technical products. Hence we must know how it has been defined in evolutionary biology, since we are going to use the same definition here. Later, we also build some further concepts, such as the adaptive fitness landscape, on the concept of fitness. Since we aim to create new contribution for usability in this research, we also have to assess the present paradigm and definitions of usability. This is done by reviewing usability literature. Finally, we define a new concept of system usability, which is intended to complement the present body of knowledge, and is particularly meant for the domain of complex technical systems.

In the context of this research, the first approximation of product evolution and market success is that the markets judge the products based on their technical *fitness*. In real life, there are also other factors, such as politics, credibility, brand, marketing etc. affecting the success of the products, but here we concentrate on technical fitness.

Technology companies produce complex technical systems, which are sold on the markets. By using the approximation mentioned above, we can say that the market success is based on fitness of the products. There are also many other factors contributing to fitness besides usability, for example ISO/IEC 9126-1 2001[1] defines six attributes of software quality, one of which is usability. The rest are: Functionality, efficiency, reliability, maintainability and portability. In addition to the software quality issues, the hardware features, such as design, speed, capacity, durability and fault tolerance are significant contributors to fitness. Even price, although not essential in high-tech products, is one factor. Usability can be considered to have the conventional HCI-oriented usability and the complex-system-oriented system usability (see Figure 2).

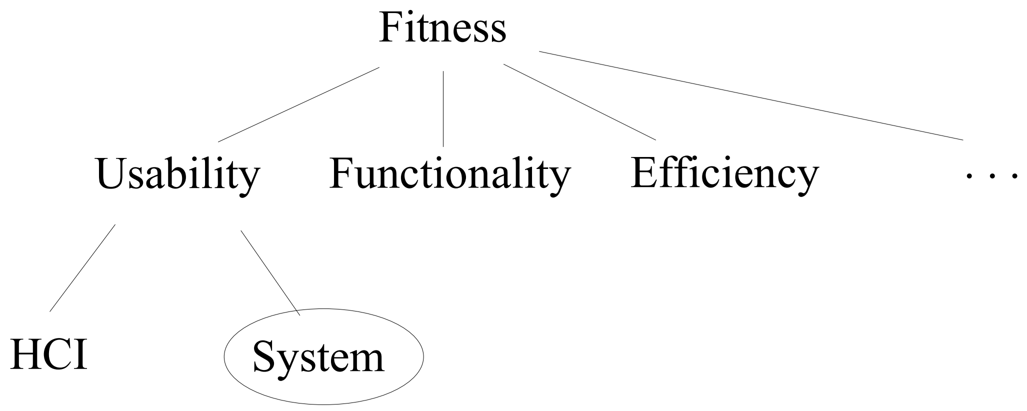


Figure 2. Usability is one contributor to fitness. The existing HCI-centred paradigm and system usability are complementary. System usability has also some overlapping with other areas, such as functionality, although from a different view point.

As the system usability has been defined with the aid of total user effort (see chapter 2.3 'System usability'), it has some overlapping with other attributes. For example, functionality is the basic bread and butter for engineers. It is functionality in particular that the designing engineers create. Many times, this is so accented that nothing else but the functional specification is considered. However, functionality is an important factor also in system usability consideration. When modifying the old functionality or creating some totally new, the effort required for a user's task might be drastically reduced. By definition, this means improvement in system usability. Also efficiency seems to belong to the same category – if efficiency is improved, it also improves system usability. Anyway, this kind of overlapping is not exceptional. Functionality and efficiency are not totally independent either, since new kind of functionality may also improve efficiency.

Since the field of this research is usability, the other areas are not further studied here. However, we shall later extend the relations of usability and complexity by introducing a new element of *user-observable complexity*. The extended view will be studied in chapter 4.3 'The relation of complexity and system usability'.

2.1 Fitness

Fitness is a concept developed in evolutionary biology. Brodie 1997a[22] tells about natural selection and fitness the following way: *Natural selection* is a process that 1) can only work with available materials (i.e. individuals that exist and the variation they represent), 2) favours variation that increases reproductive success, i.e. it does not make organisms "better" in some aesthetic sense, or make them more "fit" in any sense except in terms of reproduction, 3) works only relative to the current options (it favours the best available variants, not the best of all possible worlds). Selection acts through *fitness*, which is, differential reproductive success. Usually this is thought of as the number of offspring produced. This is absolute fitness, a measure of fitness on an absolute scale. Relative fitness is useful in comparing individuals in a population, and it is absolute fitness divided by the average absolute fitness (ibid.).

According to the Encyclopaedia Britannica (Britannica.com Inc 2001[23]), natural selection is quantified by a measure called *Darwinian fitness*, or relative fitness. Fitness in this sense is the relative probability that a hereditary characteristic will be reproduced; that is, the degree of fitness is a measure of the reproductive efficiency of the characteristic. Encarta[24] defines evolution and fitness the following way. "Natural selection sorts out the useful changes in the gene pool. When this happens, populations evolve. Beneficial new genes quickly spread through a population, because members who carry them have a greater reproductive success, or evolutionary fitness, and consequently pass the beneficial genes to more offspring". The gene pool is defined (Colby 1996[25]) so that it is the set of all genes in a species or population. According to Colby, fitness in an evolutionary sense, is the average reproductive output of a class of genetic variants in a gene pool. Thus, fit does not necessarily mean biggest, fastest or strongest. The reproductive success could be achieved either by increasing the probability of the individual to survive until the offspring are produced or increasing the number of surviving offspring. These things are not entirely independent, as the individual must also survive in order to pass more offspring.

Colby 1996[25], also tells about the relative nature of fitness. "Any organism's success depends on the behaviour of its contemporaries. For most traits or behaviours there is likely no optimal design or strategy, only contingent ones. Evolution can be like a game of paper/scissors/rock". "Occasionally, a mutation would arise that allowed its bearer to reproduce better than its contemporaries. These mutant strains would crowd out the formerly dominant strains."

Hence, we could say that fitness is a relative measure characterising the probability of survival. The relativity means that fitness of an entity is always compared to some reference – fitness of another entity. The used term 'reproductive success' actually fits the technological world quite well. If the markets judge the product to be fit, it sells well and hence the product is produced in large quantities. Based on Colby's text, we could also argue that if fitness of a product is good, the product gains market share and, in a simplified model, displaces the other products. In this research, fitness is studied with the aid of fitness landscapes[a] (see chapter 4, 'Fitness landscapes in engineering') and the NK model developed by Stuart Kauffman (Kauffman 1992[126]).

2.2 Usability

We shall now study the existing definitions of usability. This is necessary in order to understand the present paradigm and its limitations. In usability literature, the concept of usability is associated with computer systems. In this domain, the used applications are running in a computer and typically used by a group of similar type of users. The Curricula for Human-Computer Interaction (ACM SIGCHI 1992[26]) tells about the history of HCI.

Human-computer interaction arose as a field from intertwined roots in computer graphics, operating systems, human factors, ergonomics, industrial engineering, cognitive psychology, and the systems part of computer science. Work in computer graphics has continued to develop algorithms and hardware that allow the display and manipulation of ever more realistic-looking objects. Some of these building blocks include the mouse, bitmapped displays, personal computers, windows, the desktop metaphor, and point-and-click editors. Work on operating systems developed techniques for *interfacing* input/output devices, for tuning system response time to human interaction times, for multiprocessing, and for supporting windowing environments and animation. This strand of development has currently given rise to "*user interface* management systems" and "*user interface* toolkits". The problem of the human operation of computers was a natural extension of classical human factors concerns, except that the new problems had substantial cognitive, communication, and interaction aspects not previously developed in human factors, forcing a growth of human factors in these directions. Ergonomics is similar to human factors, but it arose from studies of work. Industrial engineering arose out of attempts to raise industrial productivity starting in the early years of this century. The early emphasis in industrial engineering was in the design of efficient manual methods for work, the design of specialised tools to increase productivity and reduce fatigue, and, to a lesser extent, the design of the social environment. Finally, the growth of discretionary computing and the mass personal computer and workstation markets have meant that sales of computers are more directly tied to the quality of their *interfaces* than in the past. The result has been the gradual evolution of a standardised *interface* architecture from hardware support of mice to shared window systems to "application management layers." Along with these changes, researchers and designers have begun to develop specification techniques for *user interfaces* and testing techniques for the practical production of *interfaces*.

As we can see, the approach used by ACM SIGCHI is very much computer user interface oriented and the system complexity, in design and construction of these systems, is not dealt with directly, but via the user interface. This easily leads to a situation where the usability design is neglected in many parts of a *complex technical system* and left to the designers of the user interfaces.

Mostly, the general approach in the literature is either **Human Factors** (HF) (Carey 1997[27], Chapanis 1996[28], Coe 1996[29], Norman 1990[30], Salvendy 1997[31], Sanders/McCormick 1993[32]) or **Human Computer Interaction** (HCI) (Baecker/Grudin/Buxton/Greenberg 1995[33], Casey 1998[34], Dix/Finlay/Abowd/Beale 1998[35], Helander/Landauer/Prabhu 1997[36], Karat 1991[37], Schneiderman 1998[38]). According to Curricula for Human-Computer Interaction (ACM SIGCHI 1992[26]), Human Factors study the human aspects of all designed devices and, by contrast, Human-Computer Interaction studies both the mechanism side and the human side, but of a narrower class of devices.

---

[a] The fitness landscape idea was originally introduced by Sewall Wright.

According to the same source, Human-Computer Interaction is a discipline concerned with the design, evaluation and implementation of interactive *computing* systems for human use and with the study of major phenomena surrounding them. There is also an example of an HCI-situation: "The classical situation that comes to mind is a person using an interactive graphics program on a workstation". This implies that the system is typically a concentrated computer system, which is used by similar type of users for design or surveillance duties. This description does not fit, e.g. a complex telecom network, which is a distributed system having many very different user groups.

Table 2. Content of HCI (ACM SIGCHI 1992[12])

- N The Nature of HCI
    - N1 (Meta-)Models of HCI
- U Use and Context of Computers
    - U1 Human Social Organisation and Work
    - U2 Application Areas
    - U3 Human-Machine Fit and Adaptation
- H Human Characteristics
    - H1 Human Information Processing
    - H2 Language, Communication, Interaction
    - H3 Ergonomics
- C Computer System and Interface Architecture
    - C1 Input and Output Devices
    - C2 Dialogue Techniques
    - C3 Dialogue Genre
    - C4 Computer Graphics
    - C5 Dialogue Architecture
- D Development Process
    - D1 Design Approaches
    - D2 Implementation Techniques
    - D3 Evaluation Techniques
    - D4 Example Systems and Case Studies
- P Project Presentations and Examinations

Mainly, the HCI research uses the **psychological** (Card/Moran/Newell 1983[39], Gardiner/Christie 1987[40], Norman 1991[41], Proctor/Van Zandt 1994[42]) and **sociological** (Jirotka/Goguen 1994[43], Lansdale/Ormerod 1994[44], Reeves/Nass 1996[45], Thomas 1995[46]) approach to designing usability of computer systems.

A great deal of the usability literature is dealing with **Graphical User Interfaces** (GUI) (Arlov 1997[47], Borenstein 1994[48], Fowler 1998[49], Marcus/Smilonich/Thompson 1995[50], Minasi 1994[51], Tognazzini 1992[52], Weinschenk/Jamar/Yeo 1997[53], Zetie 1995[54]) or **User Interfaces** (UI) in general (Browne 1994[55], Collins 1995[56], Cooper 1995[57], Howlett 1996[58], Mandel 1997[59], Martin/Eastman 1996[60], Mayhew 1999[61], Microsoft 1995[62]). Some books cover a subtopic of user interfaces, such as **consistency** (Nielsen 1989[63]), **graphics** (Horton 1992[64], Marcus 1991[65], Mullet/Sano 1995[66], Tufte 1997[67], Wainer 1997[68]) or **icons** (Horton 1994[69]), even **colours** (Albers 1987[70]). There are also a lot of books describing the **evaluation** and **measurement** of usability in practice (Bias/Mayhew 1994[71], Dumas/Redish 1993[72], Lindgaard 1994[73], Nielsen/Mack 1994[74], Rubin 1994[75]). Some books are **design guides** for achieving good usability when designing *computer systems*, mainly software (Beyer/Holtzblatt 1998[76], Greenbaum/Kyng 1991[77], Kyng/Mathiassen 1997[78], Landauer 1995[79], Tognazzini 1996[80]). **Style guides** define usability guidelines for specific platforms or for more general use (Apple 1992[81], Fowler/Stanwick 1995[82], Microsoft 1996[83], Open Software Foundation 1993[84], Sun Microsystems 1989[85]). There are also books for **web site** usability (Bain 1996[86], Fosythe/Grose/Ratner 1998[87], Parker 1997[88], Rosenfeld/Morville 1998[89], Sano 1996[90]).

ISO/IEC 9126-1 2001[1] divide software quality into six broad categories: Functionality, reliability, usability, efficiency, maintainability and portability. They are defined the following way:

- **Functionality** a set of attributes that bare on the existence of a set of functions and their specified properties. The functions are those that satisfy stated or implied needs.
- **Reliability:** a set of attributes that bear on the capability of software to maintain its level of performance under stated conditions for a stated period of time.
- **Usability:** a set of attributes that bear on the effort needed for use, and on the individual assessment of such use, by a stated or implied set of users.
- **Efficiency:** a set of attributes that bear on the relationship between the level of performance of the software and the amount of resources used, under stated conditions.
- **Maintainability:** set of attributes that bear on the effort needed to make specified modifications.
- **Portability:** a set of attributes that bear on the ability of software to be transferred from one environment to another.

ISO 9241-11 1998[2] defines the usability based on the degree of excellence of a product.

- **Usability:** the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use.

Within ETSI (ETSI specification ETR095 1993[13]) the ISO definition has been generally adopted. ETSI, however, enhances the definition in the following ways.

- **Usability** is considered as a pure ergonomic concept not depending on costs of providing the system. Usability together with the balance between the benefit for the user and the financial costs form the concept of *utility*. This means that an ergonomically highly usable system may have low utility for a particular user who considers the cost too high in relation to his or her need for using the system.

- **Measures of usability** are assumed to be of two kinds:

  1) Performance measures, which are "objective" measures or observation of the user behaviour and are focused on task performance.

  2) Attitude measures, which are "subjective" measures of observations of the users' opinion of working with the system.

Curiously, even though the standards, e.g. ISO/IEC 9126-1 2001[1] and ISO 9241-11 1998[2], define the usability quite comprehensively on a general level, the actual usability literature does not cover the issue quite so extensively. Perhaps the reason is that the present usability theories mainly deal with computer systems and not with highly specialised complex technical systems, and therefore the approach has been the familiar user interface-oriented. Figure 3 characterises the present approach.

According to Nigel Bevan (Bevan 1995a[91]), usability has two complementary approaches. One is a product-oriented "bottom-up" view identifying usability as ease of use. The other one is a broader "top-down" approach identifying usability as the ability to use a product for its intended purpose. Nielsen 1993[92] defines five different usability attributes in his book titled "Usability Engineering". Here also, the approach is that usability is created with a proper user interface.

- **Learnability:** The system should be easy to learn, so that the user can rapidly start getting some work done with it.

- **Efficiency:** The system should be efficient to use, so that once the user has learned the system , a high level of productivity is possible.

- **Memorability:** The system should be easy to remember, so that the casual user is able to return to the system after some period of not having used it, without having to learn everything all over again.
- **Errors:** The system should have a low error rate, so that users make few errors during the use of the system, and so that if they do make errors they can easily recover from them.
- **Satisfaction:** The system should be pleasant to use, so that users are subjectively satisfied when using it.

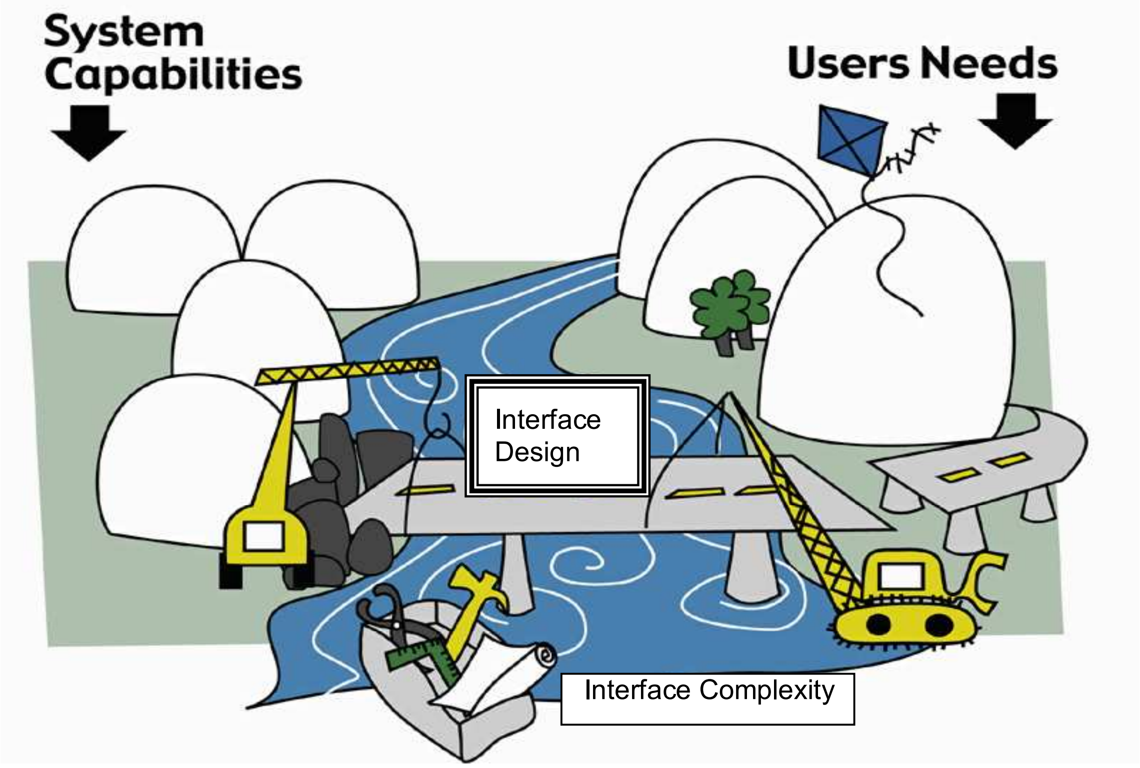


Figure 3. The present usability is concentrating on computer systems and Human Computer Interaction (HCI). There the emphasis is on the user interface. The basic idea for the picture is from Nielsen's book Usability Engineering (Nielsen 1993[91]).

Nielsen also says that "Usability is not a quality that can be spread out to cover a poor design like a thick layer of peanut butter[a], so a user-hostile interface does not get user-friendly even by the addition of a brilliant help system". This is very true. However, by going a little further and applying the system usability principles, we could say that a hostile *complex technical system* does not get user-friendly even by addition of a brilliant *user interface*.

---

[a] According to Nielsen, the peanut butter metaphor was originally by Clayton Lewis.

The other area in usability research is the simple devices. Turkka Keinonen (Keinonen 1998[93]) has studied one-dimensional usability of those. His starting point is the idea that the usability is a trade-off between ease-of-use and functionality. Based on this principle, products are either easy to use with few features, or they are versatile and difficult to use. According to Keinonen, if a product is observed to have more functions, or to be easier to use than what is expected by the one-dimensional usability rule, then it is considered a good choice.

In complex technical systems, this is not quite so. In this domain, the solution space is so huge that it is always possible, at least in principle, to find very usable solutions. However, due to the vastness of the solution space, it might be very difficult to find them. In this document, we try to find some ways to cope with complexity when building these technical systems.

2.3 System usability

We shall define the concept of system usability in this sub chapter. However, since the complexity approach is very important in system usability of complex technical systems, we shall also concentrate on complexity in chapter 3. The picture of system usability will be enhanced in chapter 4.3, where we define the relation of complexity and system usability.

System usability is built on the ISO/IEC 9126-1 2001[1] definition of usability (see above) in the sense that the standard mentions the *effort* needed for use. In complex technical systems, the complexity is scattered in many different subsystems and devices, which are not a part of a computer system in the conventional sense and do not necessarily have a direct user interface. These systems are used and maintained by a number of very different groups of personnel, some of who are sitting in the control room at computer terminals, some are working in the field, some in planning offices etc.

The existing usability theories describe the usability of simpler devices quite aptly and they can also be applied to the user interfaces of complex technical systems, but they do not provide sufficient insight alone for building very usable complex technical systems. As the system usability is defined with the aid of the total effort that all user groups together require for using the system, it provides a more comprehensive approach. System usability can be characterised with an illustration such as the one in Figure 4. Here we have to note that system usability is meant for complementing the present usability thinking, not to replace it.

If we compare Figure 4 with the picture, which is on the cover of Nielsen's book Usability Engineering (Nielsen 1993[91], Figure 3 in this text), we can see the main difference: Usability means building a proper interface between the user and the machine, whereas *system usability* means constructing the whole complex system so that the operating effort of all the different user groups is minimised. In a complex system, such as a cellular network, this comprises many things that have very little to do with computer interfaces, for example, radio network features, control algorithms, parameters, configuration etc, as well as the interdependence of the various parts and operating tasks.

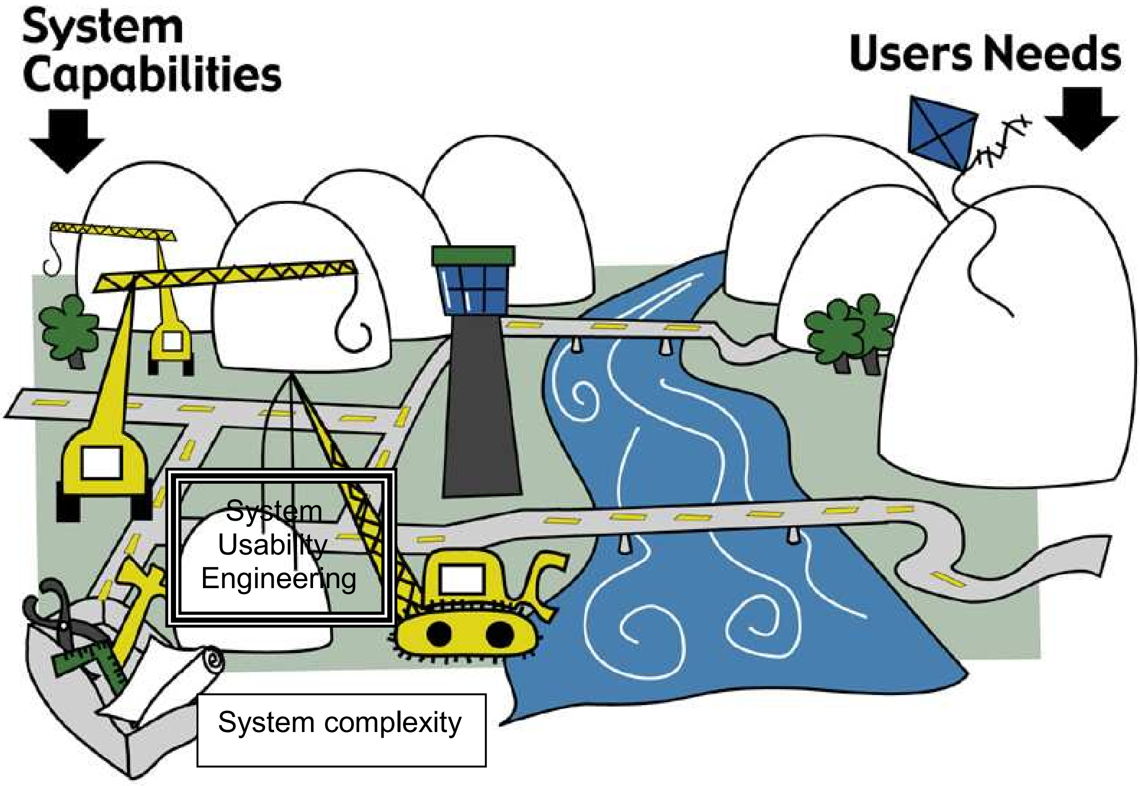


Figure 4. System usability. Here, the emphasis is on the system itself and not on the user interfaces, which are well covered by the existing usability paradigm.

As discussed before, there are some other factors besides usability that affect fitness of technical products. If we keep those other fitness contributing factors invariable, it is the changes in usability that cause the fitness fluctuations. Since we do not concentrate on the other factors in this research, we can identify fitness with usability (or system usability, more precisely) for the remainder of the document. Therefore, we do not have to define *usability* landscape in addition to *fitness* landscape (fitness landscape is introduced in chapter 4, 'Fitness landscapes in engineering').

The author has defined the system usability by means of the effort operators have to put in operating the network (see Equation 1). The total effort in system usability is divided into five different categories, which characterise the different operations:

- Planning (E_{Plan})
- Building (E_{Build})
- Using (E_{Use})
- Improving ($E_{Improve}$)
- Managing the services ($E_{ServMan}$)

The number and quality of the different categories were originally based on the researcher's experience on cellular networks, but they were also verified with the operators when doing the research. They are also named so that they could be used generally with complex technical systems. When all these efforts are combined, we can then take a reciprocal of the sum and get a figure to characterise the system usability (see Equation 1).

$$SystemUsability = \frac{1}{E_{Plan} + E_{Build} + E_{Use} + E_{Improve} + E_{ServMan}} \qquad \text{Equation 1}$$

where E_x = Effort used by operator

All the operating efforts can be listed under these categories. However, the operating tasks to be listed under these categories are not quite the same for all the operators and they may use different names for the tasks, so the list of efforts would not be uniform over the operator space. Particularly, for different kinds of complex technical systems the efforts would not be the same. This is not even necessary, since the main thing is to sum up all the operating efforts in all categories and then take a reciprocal of the sum. The different efforts can be measured, for example, with the aid of man-hours the different user groups have to invest in their tasks, amount of training that is needed to handle the necessary equipment, employee turnover rate giving an indication of how tedious the work is considered to be etc. The most straightforward way for measuring would be to measure only the man-hours, but quite obviously many nuances would be lost in that case. It is also recognised that different people have different levels of expertise. This means that some high level expert could benefit the operations in one hour more than some other people in a day. Further development could be assessed in further research. If a good set of metrics is found, the increase in system usability could be measured by checking the situation from time to time.

Since the different efforts contributing to the system usability are not of equal significance, we should have weight factors for the efforts. In this way, we can scale the factors so that the mutual relationship is approximately correct to give realistic measures of total system usability (see Equation 2). The sum of the weight factors in the equation is 1.

$$SystemUsability = \frac{1}{\alpha E_{Plan} + \beta E_{Build} + \chi E_{Use} + \delta E_{Improve} + \varepsilon E_{ServMan}} \qquad \text{Equation 2}$$

In addition to the weight factors, system usability should also be calibrated with the system size. This means that we would define the total effort scaled with, for example, number of base stations, TRXs[a] or subscribers (see Equation 3).

$$SystemUsability = \frac{(Nr)_{BTSs}}{\alpha E_{Plan} + \beta E_{Build} + \chi E_{Use} + \delta E_{Improve} + \varepsilon E_{ServMan}} \qquad \text{Equation 3}$$

We could also define *relative* system usability by dividing the absolute system usability with the number of functions in the system (see Equation 4). Then it would actually be possible to get an implication of how the number of functions affects system usability. If, for example, there is an interdependency between the various functions, then increasing the number of these interdependent functions would decrease the relative system usability and eventually lead to a complexity catastrophe (for definition of complexity catastrophe, see chapter 4.2.2).

$$SystemUsability = \frac{(Nr)_{BTSs}}{(\alpha E_{Plan} + \beta E_{Build} + \chi E_{Use} + \delta E_{Improve} + \varepsilon E_{ServMan}) * (Nr)_{func}} \qquad \text{Equation 4}$$

An interesting question would also be: What is the interdependence of the different efforts in operating a certain system? And how significant is it? It is quite obvious that the increasing K-factor (interdependence) would make the operating more difficult and slow, thus hampering the efficiency. Reduced efficiency means more effort and this way lower system usability. Perhaps the excessive interdependency in the efforts (work group tasks) would also lead to a complexity catastrophe in a large system.

In addition, if we want to compare the system usability of several completely different complex technical systems, we have to introduce one more weight factor for the system usability of every system (see Equation 5).

---

[a] Transmitter/Receiver.

$$SystemUsability = \frac{\phi(Nr)_{BTSs}}{\left(\alpha E_{Plan} + \beta E_{Build} + \chi E_{Use} + \delta E_{Improve} + \varepsilon E_{ServMan}\right) * (Nr)_{func}} \qquad \text{Equation 5}$$

Of course, system usability is also affected by operating tasks (processes). Even a system with good usability can be inefficient to use if the operating processes are poorly designed or unfit for the system (or vice versa). Therefore, we should always consider both the system usability and operating task procedures in connection with each other. That is why it would be advantageous to define the operating task procedures, for which the system usability is optimised. For any other procedures, the system usability would not necessarily be as good. Operators could then take calculated risks, if they want to deviate from those recommended procedures.

As the complexity point of view is important for system usability, we shall next concentrate on studying what complexity is, and how this understanding could be beneficially used in technological development.

3. COMPLEXITY

In order to understand complex technical systems, we must first have an understanding of complexity. The aim of this chapter is to a) Define what we mean by complexity, b) Assess how nature uses hierarchical structure, modularity and autonomous subunits in building complex systems, c) Define user-observable complexity in order to have a better understanding of user's point of view to complexity, d) Study how complexity is increasing in nature and in technology, e) Learn how to reduce user-observable complexity when creating technical systems. Here we have to note that this does not necessarily imply reducing functionality. There are ways to reduce user-observable complexity without giving up functionality. We also take a brief look at complex *adaptive* systems, which are common in nature, but are not yet widely used in technology. Of course, technology can be considered adaptive in the sense that it is constantly modified for different situations by engineers, but it is not adaptive in the sense that a certain system would autonomously evolve and adapt to changing environments. This is useful, since it enables eliminating the run-time parameter tuning effort and thus improves system usability.

3.1 What is complexity?

What is complexity and what is it that makes a system complex? We can find several characterisations in the literature. Usually the descriptions characterise complexity as an interaction of many different components and emergent features rising from the interaction.

"Whenever you look at very complicated systems in physics or biology, you generally find that the basic components and the basic laws are quite simple; the complexity arises because you have [a] great many of these simple components interacting simultaneously. The complexity is actually in the organisation – the myriad possible ways that the components of the system can interact". This is how George Cowan (Waldrop 1992[94]) describes complexity. When any complicated system is divided into small enough components, the functions of those components can be understood. However, the down side is that one cannot necessarily understand the functioning of the whole by studying the components. A system is, by definition, an entity with emergent properties, which rise from the interaction of the parts (see Bertalanffy 1984[95]), (see also the definition of Systems Theory in Appendix 2).

Kaneko and Tsuda (Kaneko/Tsuda 1994[96]) emphasise irreducibility, i.e. a complex system ceases to be complex when divided into separate pieces, and dynamic multiple inter-relations. In Poon's and

Greborgi's article (Poon/Greborgi 1995[97]), complexity has been described with three qualities: 1) A complex system is composed of many parts that are inter-related in a complicated manner. 2) A complex system possesses both ordered and random behaviours (see *effective* and *random complexity* in chapter 3.3 'User-observable complexity'). 3) A complex system often exhibits a hierarchy of structures.

Nature has developed complex systems on the earth for about $4*10^9$ years. Even a single living cell is beyond the present man-made technology, not to mention the complex multi-cellular organisms, such as a human being. The biosphere is immensely more complex than anything human has ever produced and it works very well without intervention from humans. In nature, adaptivity is a condition for life, since only the fittest will survive and poor solutions are destroyed without mercy. If we want to learn how to build usable complex technical systems, it is valuable to understand how nature is building complex systems. Therefore, the biological and complexity theories play a major role in the theoretical framework of this research.

3.1.1 Hierarchy, modularity and autonomy

Complexity researcher John Holland argues (Holland 1995[94]) that a complex system has many levels of organisation, with agents at any one level serving as the building blocks for agents at a higher level. Francis Heylighen (Heylighen 1994[98], Heylighen 1999[99]) mentions functional hierarchy of control levels and sequence of metasystem transitions (Turchin 1977[100]). He uses this expression about natural evolution emphasising the spontaneous transitions to a higher (meta-) level, which form the steps of evolution. For example, genes are combined in chromosomes, several prokaryotic cells are united in a eukaryotic cell, cells are combined to form multi-cellular organisms etc. (Schuster 1996[101]). Schuster gives an example of the transitions in nature in five steps:

- Independent replicators compete for resources optimising their own benefits
- Mutual dependence of reproductive success eliminates competition
- (Previously optimised) replicators grow together and form a functional unit
- Replicators are integrated to create a new unit of selection at the next hierarchical level
- Integration creates a new class of individuals evolving as autonomous units

For example, in a biological entity the cells are forming organs and the organs form the organism. According to Schuster (Schuster 1996[101]), nature uses three major design approaches when building complex systems: 1) Optimisation to deal with scarcity, 2) Innovation to take advantage of abundance and 3) Modular design and tinkering to master unpredictability.

Optimisation makes organisms more efficient – they become better at exploiting the available resources. Optimisation, however, does not lead to radical innovation. Innovation is created when a replication error occurs in DNA duplication and causes a piece of DNA to be copied twice. Then the second set of genes is free to develop novel functions, since it is not required for ordinary cellular life. The chances for survival of the new variant are good, when resources are cheap, since the cost of having a larger genome is negligible. Tinkering means building upon the latest version. Nature is very efficient in tinkering, but it also makes mistakes (ibid.). For example, in the vertebrate eye the nerves and blood vessels come in on the wrong side of the retina. This mistake was not made in the design of mollusc or insect eyes, which are made using the same genes in the same order. Insect eyes illustrate a design principle especially well suited for tinkering (Saura 2000[135]). This approach (using same platform) is used unless a major catastrophe destroys the existing platforms making room for major innovations and restarting the development from another point.

One principle of design that is especially well suited for tinkering is *modular construction* and it is found everywhere in nature. An enormous variety of different things can be assembled from a few modules (Schuster 1996[101]). Modularity allows the adaptation of different functions with little or no interference with other functions (Wagner 1995[102], Bonner 1988[103]). Bonner considers modularity a prerequisite for the adaptation of complex organisms.

All these are also used in technology. Optimisation usually happens between the releases of the same product, when initially constructed functionality is improved for the next release. This redesign has been noted to be a very important success factor (Brown/Eisenhardt 1998[184]). Innovation, of course, cannot be overvalued in high-tech. The product development cycles are getting shorter and shorter placing high requirements for innovation capacity of the technology companies in order to maintain continued success. Mostly, the new products follow evolutionary development, in other words, new products are based on the old ones (tinkering). To ease rapid product development and save some development effort, modular structure is utilised. Modularity addresses product complexity through decomposition of systems, partitioning of functions, analysis of interactions and modular assembly (Marshall/Leaney/Botterell 1999[104]). Design modularisation has been studied across a range of product manufacturers from automotive to scanning and sensing equipment and manufacturing machinery (see Marshall/Leaney 1999[105]).

An essential feature of the systems produced by nature is a rather high degree of autonomy of the lower level sub-units. For example, the working of the central nervous system is a hierarchic affair in which functions at the higher level do not deal directly with the ultimate structural units, but operate by activating lower patterns that have their own relatively autonomous structural unity (Weiss 1951[106], see also Koestler 1964[107]). When an organism interacts with the world, the structure of the input does not produce the structure of the output, but merely modifies the intrinsic nervous activities that have a structural organisation of their own (ibid.). The functionality is implemented in the lower level units and only activated from higher levels.

One way of managing complexity, is to organise the elements into *local* networks or modules which, because of their connectivity, have strong, well defined behavioural characteristics. This reduces the *global* burden of producing coherent behaviour, since the internal behavioural co-ordination of the modules is substantially handled locally (Christensen/Collier/Hooker 2000[172]). If we apply this to technology, it means that when building very complex technical systems, the complexity should be handled locally at the appropriate hierarchical level to avoid the huge number of links to the central control point and the required processing power of any concentrated unit for handling all the functionality of the global system. Only the tasks that need to be processed at higher levels should be elevated up there. This brings up the concept of distributed processing. Perhaps the best example of the power of distributed computing is in the Internet's domain name system (NY Times 2001[108]). Moreover, the advent of distributed computing will entail an order-of-magnitude increase in complexity compared with today's single-machine software systems (ibid.). In natural adaptation, parallel distributed information processing is used, because it enables some degree of learning without prior knowledge (in contrast of look-up table information processing systems, such as expert systems, which require huge amounts of prior knowledge to be capable of only trivial learning), and because of its fault tolerance (Christensen/Collier/Hooker 2000[172]).

A dramatic demonstration of autonomous functionality can be found in Jeffress 1951[109]. The experimenters had amputated the limbs of a salamander and reversed them before reattaching. After recovering, the animal crawled backwards whenever it wanted to crawl forwards and vice versa. This was done in a developed animal, but the same operations have been done (Jeffress 1951[109]) in embryos, and these animals have then functioned in reverse from the very beginning. In another transplant experiment (Tinbergen 1951[110]), only the two forelimbs were interchanged and reversed. The grafted limbs moved just as they would have left in their original position, causing backward motion when the rest of the animal was trying to move forward, and forward motion when the rest of the animal was trying, for example, to avoid a noxious stimulus presented in front of it. A year's experience did not change this reversed movement of the grafted legs. These experiments clearly show the high degree of autonomy on the lower hierarchical levels. The functionality offered by the lower level is only activated from the higher level, not controlled in detail by the superior level.

3.2 Complexity increase

In this sub chapter, we study how complexity is increasing in nature and also in technical systems. Although (effective) complexity is potentially a positive phenomenon, since it enables many things that are not possible for simpler entities, building complexity is a dangerous endeavour and can lead to serious problems if increased uncontrollably. Many times, the wanted functionality dictates the level of complexity and the systems are designed using the known technologies without any complexity guidelines just to fulfil the requirement specification. This has lead to a development where technical complexity increases rapidly. In complicated technical systems, the increasing complexity should always be concealed, so that the user-observable complexity is not rising and spoiling the system usability.

Unfortunately, the increasing complexity is, in many cases, very much visible to the user of these systems. A good lesson we could learn from natural systems is that complexity is never increased unnecessarily, but rather that complexity *increase* is minimised (Heylighen 1994[98], Heylighen 1999[99]). In nature, the increase of complexity advances along a kind of a saw-tooth curve (see Figure 5), where the level of complexity is occasionally reduced, although the development trend is rising.

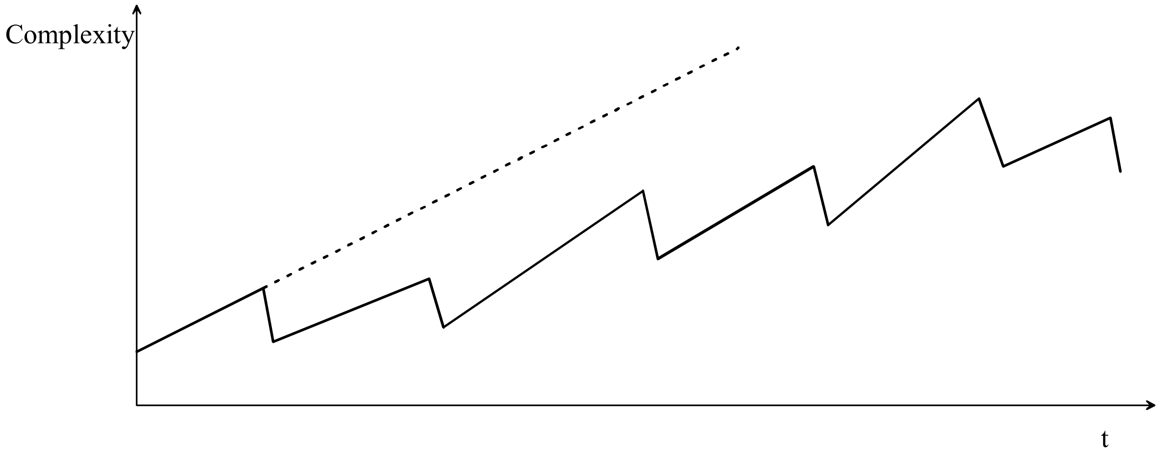


Figure 5. Evolution of complexity in nature (the basic idea e.g. in Edmonds 1995[119]). Sometimes the complexity is decreased (optimisation), but the general trend is rising. The dashed line illustrates complexity increase without optimisation.

Why does the complexity increase happen? The fundamental laws of nature are very simple, but those laws are probabilistic (Gell-Mann 2002[122]). So we have a branching tree with probabilities, with accidents at the branchings. As time goes on, the accidents can accumulate, making possible the emergence of more and more regularities. Gell-Mann speaks about frozen accidents (Gell-Mann 1995[111], Gell-Mann 2002[122]) and how they keep accumulating, as the universe grows older thereby giving more opportunities for effective complexity to increase. Hence, the envelope of complexity tends to expand, even though any given entity may either increase or decrease its complexity during a given time period. This can also be explained by coevolution (coevolution is explained in chapter 4.4) and by a principle called 'diversity begets diversity' (see Kauffman 1995[134]). To put it short, this means that agents in coevolution enhance each others fitness. On the other hand, the variety of agents increases, since the larger the number of different agents, the more there are niches where new agents can flourish. In this way, nature is conquering the 'adjacent possible' (Kauffman 2000[112]) all the time. The more experiments are made, the larger the set of adjacent possible becomes. In other words, nature is extending the set of experimented configurations indefinitely. This also applies to technology. The larger the number of different technical devices created, the larger the set of adjacent possible. This creates new niches for new innovations and the increasing diversity creates more and more diversity. Coevolution improves the fitness of the products and increases their complexity.

Frozen accidents refer to persistent phenotypic characters that are selected out of a range of possible, structurally distinctive alternatives by specific random events in the evolutionary past (Crutchfield/Nimwegen 1999[113]). For example, mutations are a completely random phenomenon – the development direction comes from selection. If some experiment turns out to be a success in the existing environment, then this solution is maintained and built upon. This way the successful design, which was more or less an accident in the first place, gets frozen and is passed on in the long chain of evolution. The solution once found, is locked in. This chain is sometimes broken by a natural catastrophe or a superior design, which also happens by chance (spontaneous order may assist, Kauffman 1992[125]). There are also analogies for these phenomena in the technology domain, where sometimes entire industries disappear and disruptive technologies displace the dominant designs. The lock-in seems to play a significant role in the complexity increase. Without them, all design would start from scratch, and obviously there would not be any complexity increase, only different variations of simple designs. It would be extremely unlikely to create a complex system all at one go without building on any existing design.

The tendency of increasing complexity can also be seen in engineering, but the complexity is not always optimised, particularly in the case of complex systems under pressure for rapid development. Sometimes, new technology is constantly built on top of the old technology without any redesign (optimisation), which increases the level of unnecessary user-observable complexity quite rapidly (dashed line in Figure 5). Particularly, in complex technical systems, the complexity of a certain technology can become quite high before a simpler way of reaching and exceeding the same goals is discovered. Sometimes, it also happens that the development of a complex system takes so much effort that there are no resources left for finding simpler ways. Many times, though, this is a symptom of a lack of lateral thinking in the management, daring to examine radically new approaches.

3.3 User-observable complexity

In this sub chapter, we introduce the concept of user-observable complexity, which characterises the user's point of view of system complexity. This is a subjective experience as opposed to the inherent complexity defined before.

Murray Gell-Mann (Gell-Mann 1994[114]) talks about *apparent complexity,* which he divides into two components: 1) random and 2) non-random complexity. In his terminology, the latter one is *effective complexity.* According to him, it is roughly the length of a concise description of the regularities of the system (Algorithmic Information Content of the regularities, see chapter 3.4 'Complex adaptive systems'). Diagnostic to regularities is the 'mutual information', which is information common to many parts. For example, the description of a regular necktie can be done by indicating the colours, widths and spacings of the stripes, as well as the background colour of the tie (Gell-Mann 2002[122]). However, soup stains and little irregularities in the weave would not be included in the description. These latter features would, in this case, represent the random complexity. The same approach can be applied to technical complexity to form the concept of user-observable complexity, which would also consist of two parts: effective and random components.

Effective complexity is the useful part giving benefit to the user. Random aspects of complexity do not contribute to anything useful, but increase the user-observable complexity. A complex adaptive system (e.g. a human being, see chapter 3.4 'Complex adaptive systems') tries to filter out the random complexity and pick up the effective complexity when interacting with the external reality. We shall study the user-observable complexity more closely in the following.

User-observable complexity can be defined as a geometrical sum of effective complexity $\overline{\mathrm{E}}$, and random complexity $\overline{\mathrm{R}}$ (see Figure 6). Increasing either one, increases user-observable complexity.

We can also define an effectiveness factor to characterise the effectiveness of user-observable complexity in a particular case. This factor is cos φ, where φ is the angle between user-observable complexity and effective complexity. The whole range of cos φ is 0…1. When the angle goes to 0, the user-observable complexity equals the effective complexity and cos φ is 1. In this case, the random complexity would be totally eliminated. When random complexity is high, the effectiveness factor (cos φ) is low.

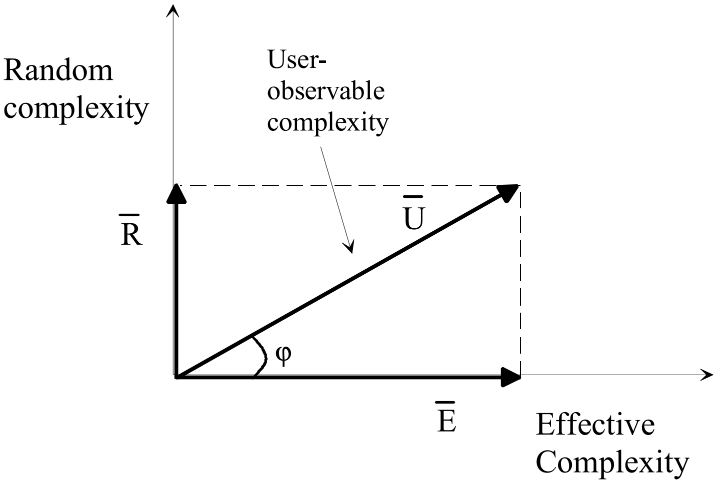


Figure 6. User-observable complexity consists of two components: Effective and random complexity. Effectiveness factor can be defined to be cos φ. In other words, if φ = 0, user-observable complexity is equal to effective complexity, and random complexity is 0.

All technical systems have these components in variable proportions, and particularly a high level of random complexity makes the system difficult for the users. This increase in difficulty can be explained by the working principle of complex adaptive systems (e.g. humans), according to which they tend to identify the effective complexity (regularities) and ignore the random complexity. The more random complexity is involved, the more difficult the task becomes. This is studied in more detail in chapter 3.4 'Complex adaptive systems'.

Tang/Salminen 2001[115] introduced the concept of complicatedness, which is very similar to user-observable complexity. According to them, complicatedness is the degree to which a decision unit for the system is able to manage the level of complexity presented by the system. This decision unit can be another system or a person. In the model, systems are designed to operate and be managed around an optimal point of complexity C* (see Figure 7). When complexity increases above C*, the decision unit can manage the system with decelerating effectiveness.

Tang & Salminen have also introduced the concept of architected complexity, which can be used to reduce complicatedness. This means taking advantage of a layered structure, where each layer presents to decision units a much less complicated system image. It is very close to the concept of metasystem transition, which will be studied later in this document. The layers should be designed so that there are mainly only intra level interactions among the elements of a layer (ibid). This kind of layered, autonomous organisation seems to be one of the main principles nature uses when building complex systems.

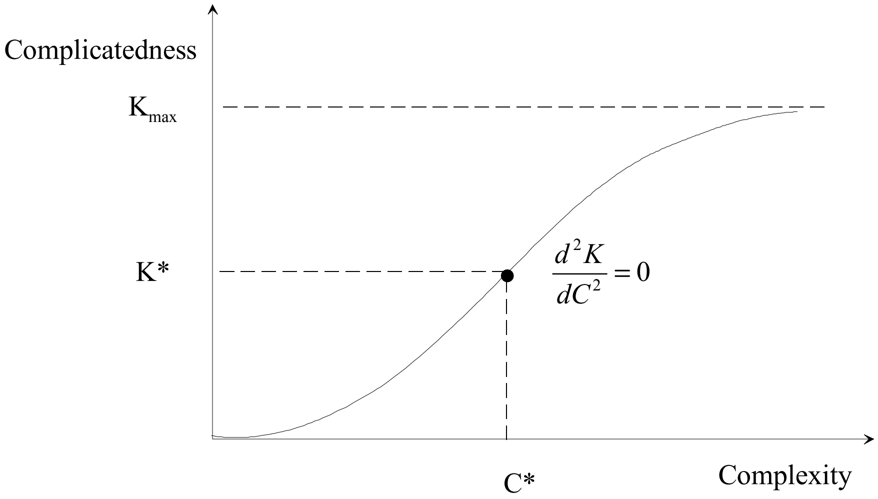


Figure 7. Systems are designed to have a certain optimal (inherent) complexity C*, which conforms to a certain level of complicatedness K*. The decision unit of the system (human or artificial) must be able to handle this complicatedness. Source: Tang/Salminen 2001[115].

In layered complex technical systems, the user-observable complexity is to be understood as the complexity observed from the level of access. For example, different user groups might use the system from different levels, and hence have a very much different user-observable complexity. In addition, when metasystem transitions are created (see chapter 3.3.1), the user-observable complexity is reduced.

To get a full view to the complexity of a technical system, not even a comprehensive *technical* scope is always sufficient, but we also have to consider the social context in connection with the technology. In other words, we should study the combination of the technology and its users, as well as the designers. Vicente (Vicente 1999[116]) recognises the following aspects in socio-technical systems to contribute to complexity:

Large problem spaces. Complex systems tend to be composed of many different elements and forces. Consequently, the number of potentially relevant factors that designers and workers need to deal with is enormous.

Social. Complex systems are usually composed of many people, who must work together to make the overall system function properly.

Heterogeneous perspectives. The workers in a complex socio-technical system frequently come from different backgrounds and thus represent the potentially conflicting values of a diverse set of disciplines.

Distributed. The demands associated with social co-ordination can be complicated by the fact that the people (and pieces of the technical system) involved may be located in different places. The fact that these people come from different cultures with diverse expectations and values can introduce additional problems.

Dynamic. Complex systems are usually dynamic and can have long time constants. It can take even days for the work domain to completely respond to an action from its workers.

Hazard. There is also a high degree of potential hazard in operating complex socio-technical systems, because inappropriate human beliefs or actions can have disastrous consequences.

Coupling. Complex socio-technical systems also tend to be composed of many subsystems that are highly coupled (i.e. interacting). This makes it very difficult to predict all the effects of an action, or to trace the implications of a disturbance, because there are many propagation paths.

Automation. Socio-technical systems also tend to be highly automated. During the abnormal situations that the automation cannot handle effectively, workers play the role of problem solvers.

Uncertainty. There tends to be uncertainty in the data that are available to the workers. Because of this, the true state of the work domain is never known with perfect certainty.

Mediated interaction. Many times the goal-relevant properties of a complex socio-technical system cannot be directly observed by unaided human perceptual systems. Therefore, computer interfaces must serve as "windows" into the system. Thus, the everyday skills that people use to routinely explore the natural environment are not sufficient to deal with the demands of mediated interaction.

Disturbances. Workers are also responsible for dealing with unanticipated events. They must improvise and adapt to the contingencies of an unanticipated event quickly to maintain system safety or productivity.

As we can see, the user-observable complexity is a rather wide topic. However, in this work, we are mainly concentrating on the issues raised by technical complexity, although some social complexity is visible in the empirical research in the form of work group interactions.

3.3.1 Reduction of user-observable complexity

In the following, we study how to reduce user-observable complexity of technical systems and why it is useful. Any process that decreases the system's total information increases one's ability to describe and control the system (Gell-Mann/Lloyd 1996[117]). Gell-Mann and Lloyd define system's total information as effective complexity + entropy term measuring the information required for describing the random aspects of the entity. In biological organisms, there is a nervous system performing this total information reduction process. The function of the organism's nervous system is to set up a symbolic model of the external world. The brain imitates or models external processes (Craik 1943[118]). This model is then used to direct the behaviour of the organism.

Any system can deal only with limited complexity at any given level, so the reduction of complexity in interaction with the environment is essential (Edmonds 1995[119]). Edmonds speaks about the complexity of the environment perceived by an organism (or institution). Consequently, we can state that reducing user-observable complexity would be beneficial. For example, Fleming/Sorenson 1999[129] have studied technological development based on patent data, and discovered that inventors appear to be able to cope with increasing numbers of components as long as those components do not interact. By definition, high number of interactions means high complexity. There will be a more detailed assessment of the implications of high complexity in chapter 4.2.2 'Complexity catastrophe'.

Some of the user-observable complexity of an implementation can be removed by hiding all but "the most relevant" details in the user interface. However, no interface is suitable for all users, since each user has a slightly different view of what is most relevant. In addition, the user interfaces can be very difficult to design, if the complexity of the system itself is unnecessarily high. Sometimes, there is no user interface at all, when the users have to deal with system complexity. For example, when radio network planners contemplate how to tune up a network, there is no user interface until the actual parameter changes are implemented. A model of the complex phenomenon must be kept inside the planners head. Thus, the reduction of the user-observable complexity should be a design principle throughout the system, not only in the user interface design.

The first possibility of controlling the complexity increase is reducing the random complexity (as opposed to non-random or effective complexity) of the technology. To get an impression of random complexity, we can study a hypothetical example. Let us assume a monkey typing text with a type writer. Obviously, the outcome is a completely random character string. In this case, the random complexity of the piece of text is very high and the effective complexity is zero. Now, if a human being edits this text, so that it becomes intelligible, the random complexity is reduced and the effective complexity becomes high.

The second possibility is to reduce the effective complexity. One example of reduction in effective complexity is the jet engine (Arthur 1993[120]), the principle of which is much simpler than in the piston engine, although the former outperforms the latter in most applications. However, the jet engine was not developed by optimising the piston engine in a stepwise manner, but instead involved more radical innovation. The third possibility is metasystem transitions, where we create a new metalevel and hide some of the complexity from the users by architecture. This actually allows us to reduce the user-observable complexity below effective complexity of the whole system.

The mentioned metasystem transition approach can be, and has been, used also in engineering. For example, the integrated circuits are easy-to-use elements, which replace complicated networks realised with discrete components. This approach radically reduces the apparent complexity of the system and eliminates the low-level design, because it is no longer necessary to deal with the details. Of course, it also limits the designers freedom, but this is not detrimental as long as the selection of circuits is not very small (see Fleming/Sorenson 1999[129]). By introducing a higher level of control and hiding the lower levels by architecture every time when the user-observable complexity reaches a certain limit, the observed complexity will not rise excessively and the user (and also the designer) can still handle the system. However, it is also possible that in the future the designs are grown without any manual designing effort (see Shipman/Shackleton/Harvey 2000[131]).

When metasystem transitions (see also architected complexity, Tang/Salminen 2001[115]) are used, the new metalevel 'isolates' the increasing complexity from showing to the users. In a metasystem transition, the increasing user-observable complexity is suddenly reduced to a lower level (see Figure 8). As the user-observable complexity tends to increase, consecutive metasystem transitions can keep it within bearable limits. A large scale example of this development is the way how computers have been used in the history. The first computers were programmed and accessed directly by using machine code. Later on, Assembly language was taken into use. Then came the first command based operating systems, and the present way of using a computer is via graphical user interfaces. Without these transitions, this kind of complexity increase would not have been possible. The metasystem transition aspect is important in *system usability* of any complex technical system. The systems should be organised in a layered structure and when the user-observable complexity increases, allow the user a new, higher level of control via automated lower levels, thus sparing the user from the very tedious work with the numerous details.

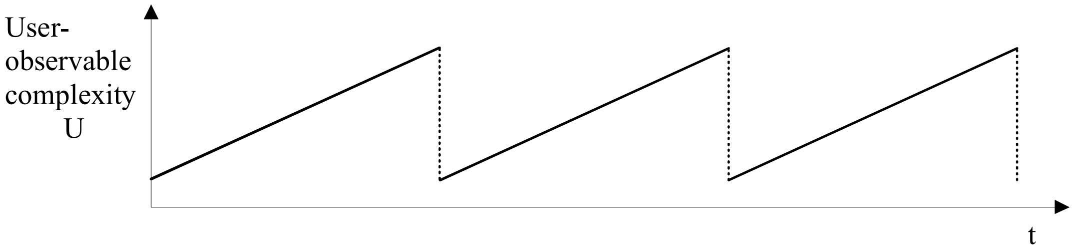


Figure 8. Series of metasystem transitions reducing user-observable complexity. Refer to Figure 5.

When following the metasystem transition principle, one should maintain a rather high level of autonomy of the lower layers in order to isolate this complexity from the higher layers. In addition, the layered and modular structure helps the product development (see chapter 4.2.2 'Complexity catastrophe'), since it is easier to design complex systems, if every change does not generate many other alterations elsewhere (Ulrich 1995[173]). In addition, one of the fundamental solutions to achieve the structured plasticity necessary for adaptability, lies in modular, hierarchical and quasi-hierarchical organisation (Christensen/Collier/Hooker 2000[172]).

3.4 Complex adaptive systems

In this sub chapter, we briefly describe complex adaptive systems, since this principle could be utilised in complex technical systems to improve system usability. The technical systems of today are mainly not *adaptive* systems, since all the possible reactions are pre-coded in the system. This also means that the users must deal with the complexity of the system every day when using and optimising it by selecting proper parameter values from the pre-coded possibilities. However, adaptive systems have a great advantage over the non-adaptive technology. The advantage is that significant amount of manual effort, which is usually required for constantly optimising the system, can be avoided due to the self-adaptation. This would radically change the system usability of these systems, since the manual optimisation could be entirely eradicated, as the system would autonomously adapt to different conditions. Thus, if we could create, for example, adaptive radio networks, the system usability of these networks would take a major leap forward. As it is quite possible that the technological development will follow this path in the future, a brief look at complex adaptive systems also enhances the applicability of the framework for further research. We shall start with some definitions.

Adaptability is characterised as the capacity to engage in interaction with the environment, thus generating new information in the system (Christensen/Collier/Hooker 2000[172]). Evolutionary adaptiveness (one of the three kinds of adaptability defined by Christensen et al.) is most commonly found at the species/population level. Evolutionary adaptability concerns the adaptability of a population to find and remain at fitness peaks in its possibility space (ibid.). Fitness peaks and landscapes are studied later in chapter 4 'Fitness landscapes in engineering.

Adaptation is considered to be a process of modifying the parameters or the structure of the system and the control actions (Tsypkin 1971[121]). The current information is used to obtain a definite (usually optimal) state of the system when the operating conditions are uncertain and time-varying. The most characteristic feature of adaptation is an accumulation and slow usage of the current information to eliminate the uncertainty due to insufficient a priori information and for the purpose of optimising a certain selected performance index (ibid.).

In order to understand the principle of complex adaptive systems, we shall now study *Algorithmic Information Content* (AIC). It is defined for a string of bits, or an entity described by that string of bits. The algorithmic information content is the length of a shortest program that will cause a given universal computer to print out that bit string and then stop computing (Gell-Mann 2002[122]). This is one way of discussing information compressing, which the complex adaptive systems make use of. When describing the entity, first it is necessary to specify the level of detail up to which the reality is described, with finer details being ignored. In other words, the data is '*coarse grained*' (ibid.).

For an incompressible random string with no regularities, Algorithmic Information Content would be very high (see Figure 9), as the shortest program to describe it would be a print command followed by the entire bit string. The effective complexity would still be small, because there are no regularities. At the other end of the scale, where the bit string is entirely regular, for example consisting only of 1's, the Algorithmic Information Content is near zero. Then the effective complexity is also close to zero, since the message 'all 1s' is so short. Hence, the effective complexity of a system varies with AIC, attaining high values only in the intermediate region between excessive order and excessive disorder (Gell-Mann 1994[114]). In other words, the effective complexity is the algorithmic information content of the regularities.

Curiously, the mentioned intermediate region between order and disorder (K is between 0 and N-1) is the domain, where natural complex systems are operating (Kauffman 1992[126]). There is a more detailed contemplation of the K-factor in chapter 4 'Fitness landscapes in engineering'.

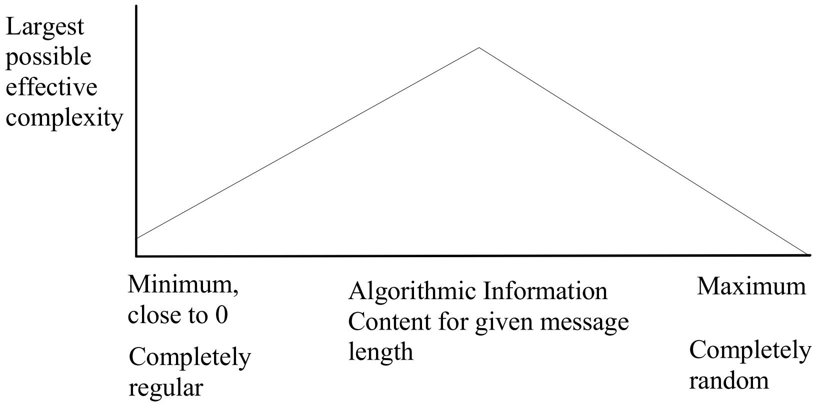


Figure 9. Largest possible effective complexity varies with AIC (Gell-Mann 1994[114]). Effective complexity is small with very regular and random data.

Gell-Mann (Gell-Mann 1994[114]) gives a description of the behaviour of *complex adaptive systems* (human, for example, see Figure 10). A complex adaptive system uses 'schemata' as a model of the external world and to make predictions. According to Bartlett 1961[123], the schemata are living, constantly developing, affected by every bit of incoming sensational experience of a given kind. Gell-Mann (Gell-Mann 1994[114]) uses the term 'schema' meaning the identified and compressed regularities that the complex adaptive system has abstracted from the available data, in other words it refers to the effective complexity.

We can study Figure 10 with an example, where a person is shown a sequence of images constituting a data stream. In the example, the person when studying the series of images, creates and continually modifies rules that are supposed to describe the regularities governing the sequence. The person combines a tentative schema based on the past with the information provided by the next few images, thus making a prediction of what will be shown later. If the predicting schema contradicts with the succeeding images, it is discarded by the person. Correctly predicting schema is retained and assigned a high value (ibid.).

If there are no regularities in the system being described, as in the case of a passage of text typed by the proverbial monkeys, a properly operating complex adaptive system would be unable to find any schema, since a schema summarises regularities and there is none. The complex adaptive system would assign zero effective complexity to the random string it is studying.

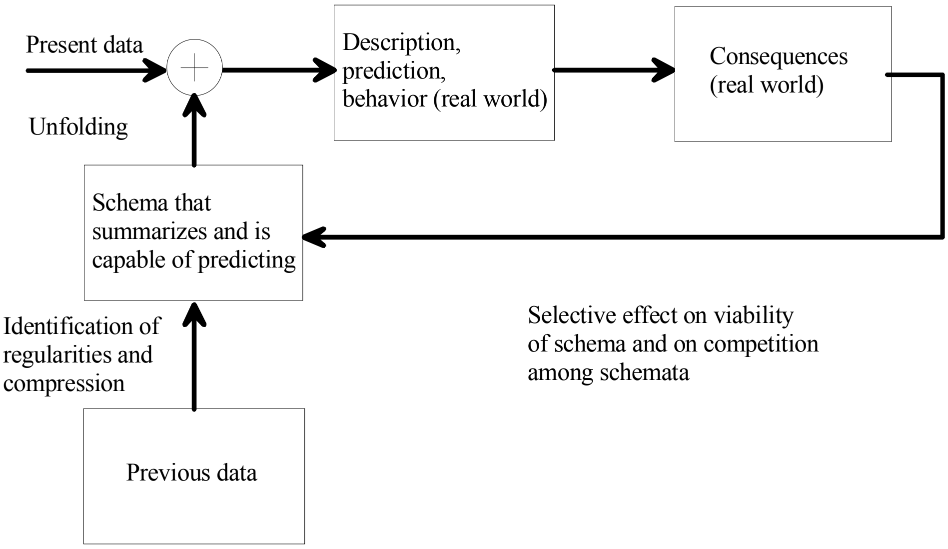


Figure 10. Functioning of a complex adaptive system. For example, a human being who is shown a series of pictures, modifies the accumulated information with the new information to produce a prediction of what is coming next. The principle is from Gell-Mann's book (Gell-Mann 1994[114]).

Complex adaptive systems tend to identify the perceived regularities of the environment and then compress the description. Applying this to humans and usability domain, we can say that a user (complex adaptive system) when using the complex technical system, tends to reduce the total user-observable complexity by identifying the effective complexity (regularities) and ignoring the random complexity. The more random complexity is present, the more difficult this task becomes. Large amount of effective complexity can also make the use more difficult, unless it is concealed in the autonomous lower layers of the system. Consequently, both the random and effective complexity should be considered in the design of complex technical systems.

Holland (Holland 1995[124]) has also studied complex adaptive systems and defines four properties and three mechanisms that are common to all complex adaptive systems. The properties are:

- **Aggregation.** Complex adaptive systems simplify other complex systems by aggregating them into categories. They can also create totally new scenes by recombining the categories, e.g. we have created the centaur and chimera. In addition, complex large-scale behaviours emerge from the aggregate interactions, such as the coherence and persistence of a large city. Aggregates can in turn act as agents at a higher level – meta-agents. By repeating this process several times, we get the hierarchical organisation, which is typical for complex adaptive systems.

- **Nonlinearity.** The whole does not equal the sum of the parts and hence linear approaches do not work with complex adaptive systems. Non-linear interactions make the behaviour of the aggregate more complicated than would be predicted by summing or averaging.

- **Flows.** Complex adaptive systems contain flows over a dynamic network of nodes and connectors. The nodes can be factories, and the connectors transport routes for the flow of goods. In general terms, the nodes are processors – *agents* – and the connectors designate the possible interactions. Tags are used to define the network by delimiting the critical interactions. Flows involve two different effects, the *multiplier effect* and the *recycling effect*. The former one means that the initial effect is multiplied by a factor when total effect passes through the network. Recycling produces more resource at each node with the same raw input.

- **Diversity.** The diversity observed in complex adaptive systems is the product of progressive adaptations. Each new adaptation opens the possibility for other interactions and new niches. Agents that participate in cyclic flows cause the system to retain resources (recycling). The resources so retained can be further exploited, thus offering new niches to be exploited by new kinds of agents. It is a process that leads to increasing diversity through increasing recycling.

The mechanisms are:

- **Tagging.** Tags facilitate selective interaction in complex adaptive systems. They allow agents to select among agents or objects that would otherwise be indistinguishable. Well-established tag-based interactions provide a sound basis for filtering, specialisation and co-operation. This, in turn, leads to the emergence of meta-agents and organisations that persist although their components are continually changing. The agent/meta-agent/meta-meta-agent/… organisation is common in complex adaptive systems.

- **Internal models.** Another hallmark of complex adaptive systems is anticipation (refer to Gell-Mann's model of complex adaptive systems, schema that is capable of predicting). The agent selects patterns in the input it receives and then converts those patterns into changes in its internal structure. Finally, the changes in structure, the model, must enable the agent to anticipate the consequences that follow when the pattern (or something similar) is encountered again. Internal models can be tacit or overt. A tacit internal model prescribes a current action under an implicit prediction of some future state, as in the case of the bacterium. An overt internal model is used as a basis for explicit explorations of alternatives, as in chess game. Both models are found in all kinds of complex adaptive systems.

- **Building blocks.** By reusing the same building blocks, we can have repetition while being confronted with perpetually novel scenes. If I encounter a flat tyre while driving a red Corvette on the highway, I immediately come up with a set of plausible actions, although I have never encountered this situation before. The use of building blocks to generate internal models is a pervasive feature of complex adaptive systems.

These are the basic properties and mechanisms of complex adaptive systems and they can also be used in technology to achieve the same kind of characteristics. This kind of technology could be used to eliminate the complicated and demanding parameter optimisation and thus improve system usability of complex technical systems. In the present paradigm of controlling, the system software is pre-coded to contain parameters, which can be used to manually change the run-time behaviour of the system. The numerous parameters – particularly interdependent ones – make the optimisation difficult and require a lot of manual effort. Complex adaptive systems have the capability to engage in interaction with the environment, thus generating new information. This capability can be considered as an ability to find fitness peaks in the possibility space (adaptive landscape) and remain at the peaks (the adaptive landscapes are described in the next chapter). Hence, it would no longer be necessary to a) pre-code all conceivable possibilities, b) manually optimise the parameters when running the system.

4. FITNESS LANDSCAPES IN ENGINEERING

In this chapter, we introduce the concept of fitness landscapes. The fitness landscape idea has been borrowed from biology (Wright 1932[125], Kauffman 1992[126]) and applied to engineering in this research. Technology development resembles natural evolution in many ways (Vincenti 1990[127], Ziman 2000[128]) and the (adaptive) fitness landscape model describes the phenomenon aptly (see Fleming/Sorenson 1999[129]). The fitness landscape concept is applied here, since it can be used for visualising evolutionary development of technology. We also explain the NK model, which can be used for defining landscapes of different ruggedness by adjusting two parameters (N and K). Then we explain how certain kinds of conditions in the landscape lead to error and complexity catastrophes. At this stage, we are ready to define the relation of complexity and usability. Finally, we use the dynamic fitness landscapes to explain coevolution of different agents in order to study coevolution of several technology suppliers and operators when developing complex technical systems.

Stuart Kauffman forms the adaptive landscape[a] such that he considers the distribution of a certain well-defined (beneficial) property, for example, the catalysing velocity of a protein, and then defines that property as the fitness of the protein. The distribution of the velocities across the space of proteins then constitutes the fitness landscape with respect to that defined function. Adaptive evolution with respect to this function is a search across the possibility space attempting to maximise the capacity to catalyse that specific reaction. However, this does not necessarily mean that the optimisation of any specific reaction would optimise the overall fitness of the organism. As an example, we could study a basket ball team. In this example, evolution would optimise fitness of the team by increasing the average length of the players. Anyway, the overall fitness of the whole team would not necessarily be optimised by maximising only one property (length). In addition, if the multiple properties are cross-connected, so that increase in one of them causes a change in the other ones, the overall fitness would probably not be at the maximum of any individual property.

In complex natural systems, there are a huge number of different traits, which in living organisms are represented by genes. The adaptive walks to better fitness most often occur through mutations of one gene at the time. Kauffman uses cubes for visualising the phenomenon (see Figure 11), and the adaptive walks are done along the vertices of the cube. This is a Boolean model, meaning that each trait can only have two different possibilities (alleles). For example, 16 (=2^4) different possibilities can be coded with four bits (traits), or we could say that four traits span a possibility space of 16 different possibilities. In the example, in Figure 11, which illustrates this kind of landscape model

---

[a] The original idea by Sewall Wright (1932[125]).

for four traits, there are three local maxima (dashed rings). All different paths lead to one of these local peaks.

The down side of this three-dimensional model, or any three-dimensional model for that matter, is that it is very difficult to visualise more than four different alternatives at each location. Certainly, this does not hint that the alternatives would be limited to this number in the real world. It only means that our capabilities to visually describe such a hyperspace are limited.

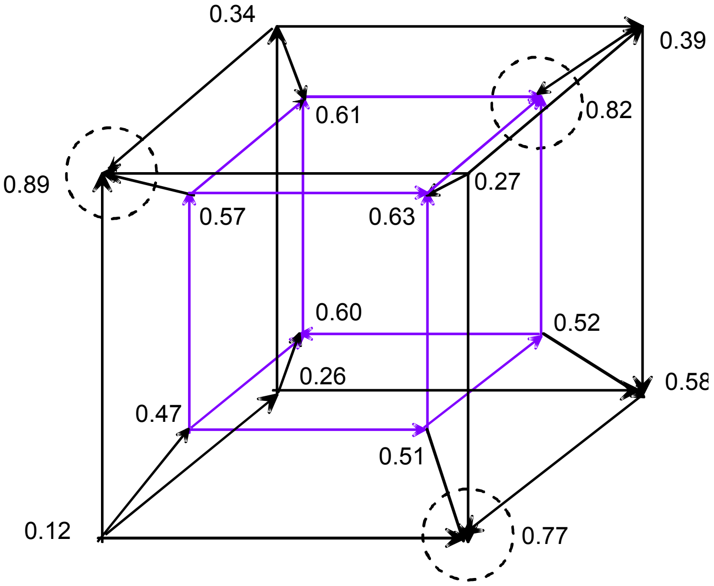


Figure 11. A 4-dimensional hypercube to demonstrate adaptive walks in a system with four bits (Kauffman 1992[125]). The adaptive walks, consisting of one bit mutations along the arrows from vertex to vertex, end up to the local peaks (dashed rings).

In the applied technical model, natural mutations are replaced by variations created by product development and natural selection by market feedback. Mutations in nature are completely random, but one could argue that the product development in industry is working in a more ordered manner. Indeed, Darwin explains biological evolution with the aid of mutations ("sports" and variation) and selection only, but in Kauffman's theory (Kauffman 1992[125]) there is also spontaneous order rising from nature, and selection more or less only refines the spontaneously rising initial solutions. In the similar manner, the product development of a technology enterprise is also able to spontaneously create initial solutions, which are then evolving with the aid of market feedback. These initial solutions are either created accidentally or drawn upon some previous knowledge (tinkering). An example of the former one is the X-rays discovered by Roentgen in 1895 (Ziman 2000[127]). In addition, the continuing growth of knowledge is effectively blind, with ideas occurring to someone in unpredictable fashion. Consequently, the analogy of biological evolution and engineering does not seem so far-fetched. For an extensive discussion of technological evolution, see Ziman 2000[127].

The different possible combinations of properties (or parts) in the complex technical system form the potential solution space from which the company must select the actual realisations when constructing the product. Selecting some combination of traits from the solution space will place the product to a certain position in the fitness landscape. The idea is to seek for high fitness, i.e. the highest hilltops in the landscape by selecting different trait combinations.

4.1 The NK model in engineering

Although technical systems are developed by engineers instead of natural evolution, the NK model can be applied to technological evolution (see Fleming/Sorenson 1999[129] and Fleming/Sorenson 2000[130]). The NK model describes striving for fitness as a *hill-climbing exercise*, where the steps are taken by creating small and large alterations (mutations) to the technical products. The alterations are selected from a solution space, which is vast in the case of complex technical systems due to the huge number of different possible combinations. Then these alterations are tested on the markets to get feedback (selection) for product development. In complex hi-tech products, it is also problematic to initially create a product with high fitness, as the market feedback is received retrospectively. This means that the 'technology push' from the suppliers is in a significant role in creating potentially fit products. To improve their chances of succeeding, they should have means to market test technology alternatives, e.g. via the Internet, already in the product development phase before the products exist. Since the adaptive landscape is dynamic, it also requires constant effort to track the changes and keep the product on the highest fitness hills. These phenomena are explained in more details in the coming chapters. In addition, improvements in the *processes* can have a significant effect on fitness. For example, if the product development process is suddenly redesigned to better suit the dynamics of the business in question, this will enable better tracking of the dynamic fitness landscape (see chapter 4.5 'Dynamic adaptive landscapes').

The fitness landscape is formed by distributions of certain traits across the system feature space. If we have two traits only, then it is quite easy to imagine what the landscape looks like, since it is a three-dimensional structure, fitness being the third dimension. However, if there are more than those two traits to consider, it is no longer a three-dimensional landscape, but a difficult-to-imagine landscape in n-dimensional hyperspace. In technical systems, the traits can be created either with hardware or with software. Usually though, the hardware only forms a platform, on top of which the features are designed with the aid of software.

In Kauffman's NK model, the complexity of an organism is measured by the number of its genes (N). Gene is a hereditary unit that can be passed on unaltered for many generations (Colby 1996[25]). There is also a cross-coupling factor (K) in the NK model, that describes how many cross-connections there are between the genes. This means that if two genes are coupled together, change in one of them will cause a change in the other one. The method of defining complexity by using the number of genes (N) may be reasonable in *living organisms* having separate genotypes[a] and phenotypes[b], but since technical products do not have genotypes (not yet anyway, although in the future this might be reality, see Shipman & al 2000[131]), we have to reconsider this definition. In technology, a more suitable definition would be that N, which is the number of parts or traits, is a 'magnitude' or a 'system size' and K is the complexity (see Figure 12 and Figure 13). This is more suitable, because a great number of parts in a technical system does not necessarily mean that the system is complex. The parts could be similar and without any coupling to each other. It is the interconnection between the numerous parts that raises the complexity. This is actually well compliant with the definitions of complexity (refer to chapter 3.1 'What is complexity?').

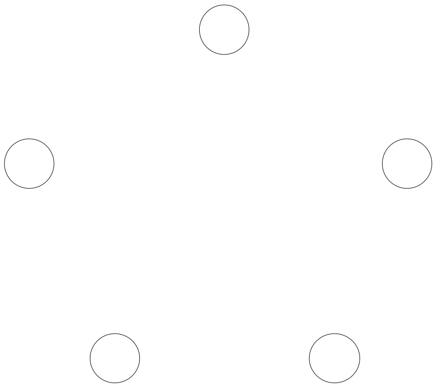


Figure 12. N=5, K=0. The traits are independent from each other. In this case, the landscape becomes smooth and single peaked (see Figure 14). The system can be large, but not complex.

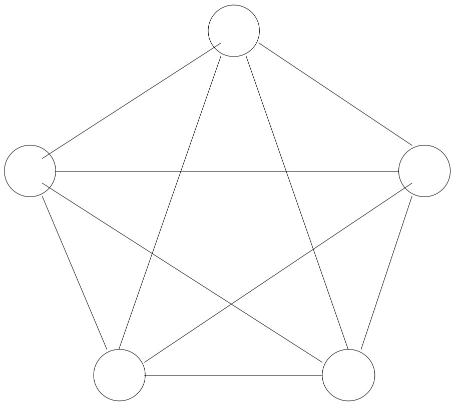


Figure 13. N=5, K=N-1. The traits are maximally connected. In this case, the landscape becomes rugged and multi-peaked (see Figure 15). The system becomes complex, when N increases.

---

[a] The genotype is the "internally coded, inheritable information" carried by all living organisms. This stored information is used as a "blueprint" or set of instructions for building and maintaining a living creature.

[b] The phenotype is the "outward, physical manifestation" of the organism.

Let us now study the N and K in the model. Each trait makes a fitness contribution, which depends on that trait and on K other traits among the N. To be able to use Boolean variables, we assume that each of the N traits can only have two possible values, 0 and 1.

In the following example, fitness is demonstrated by a calculation. The purpose of this demonstration is to show that in the non-coupled case (K=0), the changing trait does not change the other traits, whereas in the maximally coupled case (K=N-1) the change in one trait always changes all traits via the internal couplings. In this calculation, the individual fitness values (w) are assigned at random from the uniform interval between 0.0 and 1.0. Then the overall fitness (W) is produced as an average of all contributing individual factors (w). This is not the only possible way of calculating the overall fitness, other meaningful ways can also be used.

In Table 3, N=3 (not 5 as in Figure 12) and K=0, meaning that there is no coupling between the three features and consequently, when one of the traits changes, it does not cause any change in the others. In Table 4, N=3 and K=N-1, meaning that we have three different traits, which are maximally coupled with each other. The K-factor affects so that if there is some coupling between the features (K>0), then changing one trait causes a change also in the other coupled traits. However, the overall fitness is still calculated the same way as in the former case.

Table 3. N=3, K=0. Traits are independent of each other.

1 2 3	w1	w2	w3	$W=\frac{1}{N}\sum_{i=1}^{N} w_i$
0 0 0	0.6	0.3	0.5	0.47
0 0 1	0.6	0.3	0.2	0.37
0 1 0	0.6	0.4	0.5	0.50
0 1 1	0.6	0.4	0.2	0.40
1 0 0	0.7	0.3	0.5	0.50
1 0 1	0.7	0.3	0.2	0.40
1 1 0	0.7	0.4	0.5	0.53
1 1 1	0.7	0.4	0.2	0.43

Table 4. N=3, K=N-1. Traits are maximally connected.

1 2 3	w1	w2	w3	$W=\frac{1}{N}\sum_{i=1}^{N} w_i$
0 0 0	0.6	0.3	0.5	0.47
0 0 1	0.1	0.5	0.9	0.50
0 1 0	0.4	0.8	0.1	0.43
0 1 1	0.3	0.5	0.8	0.53
1 0 0	0.9	0.9	0.7	0.83
1 0 1	0.7	0.2	0.3	0.40
1 1 0	0.6	0.7	0.6	0.63
1 1 1	0.7	0.9	0.5	0.70

As the variables of the NK model are changed, it generates a family of increasingly rugged multi-peaked landscapes. In the simplest case, where K=0, the landscape becomes very smooth, meaning that the neighbouring points in the fitness landscape have nearly the same fitness value (see Figure 14). When K increases, the landscape becomes extremely rugged with such a huge number of local optima that trapping on such optima becomes inevitable. When applied to technological landscapes, this will eventually happen to a firm when it is searching through its solution space (Kauffman/-Lobo/Macready 1998[132]). Kauffman (Kauffman 1992[125]) has calculated many different key figures, which characterise these landscapes. For example, the number of local optima in the maximally rugged landscape (K=N-1) if only one-bit-mutations are used, is:

$$M1 = 2^{N}/(N+1) \qquad \text{Equation 6}$$

In the maximally rugged landscape, the fitness values are extremely uncorrelated (see Figure 15, actually there is still some correlation in this picture). In other words, the fitness value of any point contains very little information about fitness of the neighbouring points.

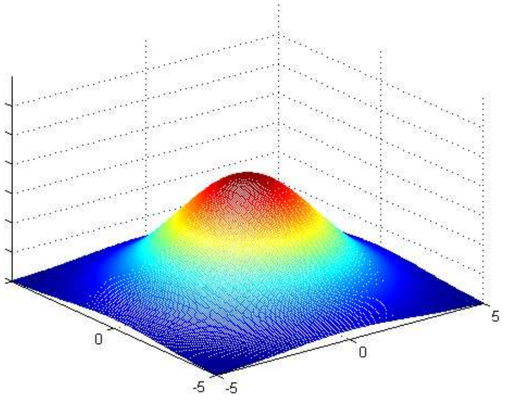


Figure 14. Independent traits (K=0) produce a smooth, single peaked landscape.

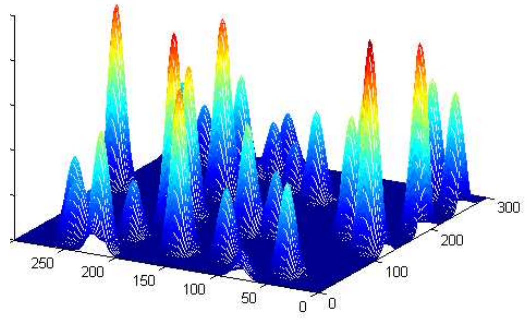


Figure 15. Cross-coupled traits (K=large) produce a rugged, multi-peaked landscape.

In natural systems, the K-value self-adjusts to a rather narrow area between 0 and N-1 (Kauffman 1992[126]). The optimal working area depends on the circumstances. Since the organisms live in coevolution (see chapter 4.4 'Coevoluton') with each other, this interaction controls the landscape characteristics and tunes the K-value to the appropriate range. For example, when selection is too low, the K-value increases causing the landscape to become more rugged, which means that the hills become steeper and hence the selection becomes stronger. If K is increased excessively, the increasing number of conflicting constraints drives the system to a complexity catastrophe (see chapter 4.2.2 'Complexity catastrophe'). This phenomenon in technology was verified by Fleming/Sorenson 1999[129], see Figure 18).

It would certainly be beneficial if the landscape of a technical product could somehow be charted and visualised prior to market exposure, but presently there seems to be no way of doing that, although Kauffman (Kauffman/Macready 1995[133], Kauffman 1995[134]) is anticipating some means for it. The difficulty is that the fitness of any product cannot be known for sure before testing it on the markets and this way forming the actual landscape. In such a case, where a hi-tech company could see the adaptive landscape, it would be possible to aim towards some high fitness hills by constructing complex technical products leading to that position.

In nature, an animal population is made up of individuals carrying a load consisting of less than optimal forms of many traits. The load may not be understood as necessarily evil; it constitutes the price of evolving that a successful population may carry with relative ease. If all individuals have nothing but the optimal trait, the population has little hope for future as circumstances change (Saura 2000[135]). We can also see the same phenomenon in technology. Since we usually do not know the fitness of the products, particularly prior to any market exposure, we have to develop different kinds of technical solutions (products) and see which of them are successful. Developing only one technology can be very risky – the fitness may not be as high as expected, and if the dynamic landscape suddenly changes, it could take a too long time to develop a new technology from scratch.

So far, we have concentrated on the evolution by 1-bit mutations (Hamming-distance =1), but that is not the only possibility. Nature also uses longer jumps by mutating more than one bit at the time (Hamming-distance >1). However, since the number of tries to find improvement after each successful step increases by a constant fraction, the rate of finding fitter variants slows exponentially (universal law of long-jump adaptation, Kauffman 1992[126]). The slowing is even faster when the conflicting constraints are higher (large K), and the landscape is more rugged.

Since the practical landscapes are somewhat correlated, it is not possible to reach very high improvements (or deteriorations for that matter) by searching close to the current location (see Kauffman/Lobo/Macready 1998[132]). This means that initially one has to jump over the correlation length of the landscape (long jumps). Later, when relatively high fitness has been achieved, the situation changes due to the exponential slowing of the success rate in long jumps. Then the probability of finding still higher peaks is greatest in the close proximity (short jumps). This seems to be the explanation why nature (and technical evolution) is using the following tactics. If search efficiency is to be maximised, the jump length is varied depending on the current fitness. When fitness is low or average, the best search distance is long, and as fitness increases the optimal search distance becomes short (1-bit mutations) (ibid.). For example, when the car was invented, many different power supplies were tried until the combustion engine settled as the paradigm and the development ever since has just been improving that.

Based on his studies, Kauffman (Kauffman 1995[134]) has also suggested that in technology the greatest diversification comes first and is then followed by more conservative experimentation. In early stages after a major innovation (a long jump) many strong variants of the innovation are tried until a few dominant designs are chosen and the rest go extinct. Later on, smaller and smaller improvements (short jumps) to the dominant designs are created until the whole theme is exhausted.

What are the economic implications of this? According to Kauffman's theory, after major innovation, there can be an early period of increasing returns, which is created in the following way. A given investment in the new technology increases productivity. As nearby products and technologies are called forth by the deforming landscapes (=coevolution, see chapter 4.6 'Coevolving systems'), each generates a new burst of rapid learning, and the consequent burst of increasing returns attracts capital thus driving further growth. Later, as improvement slows exponentially, further investments face diminishing returns and the situation cools down. As the markets saturate, further growth awaits a burst of fundamental innovation in some other sector.

4.2 Catastrophes in adaptive landscapes

To gain further understanding of complexity in technical systems and risks related to the product development of those systems, we shall now study the impact of increasing system size (number of traits) with the aid of the NK model in two extreme cases: a) The traits are independent (K=0) and hence the adaptive landscape is smooth, and b) The traits are maximally connected (K=N-1) and hence the adaptive landscape is maximally rugged. In the first case, as the magnitude (N) increases, the system gives rise to the *error catastrophe* and in the second case we are facing the *complexity catastrophe*. The ideas for error and complexity catastrophes in biology have been taken from Kauffman 1992[125]. In biology, however, the definition of N is different, since it is defined by using the number of genes in the genotype.

4.2.1 Error catastrophe

In this chapter, we shall study the situation in a static landscape, where only one kind of error catastrophe is possible. In dynamic landscapes, there are two different possibilities for error catastrophe (see chapter 4.5 'Dynamic adaptive landscapes').

In a smooth landscape, the selective force is relatively weak, since the fitness values of adjacent points are very similar, thus preventing efficient selection. In the business model, this is represented by the situation, where the different system realisations are so similar that the markets cannot see any difference between them (technology aspect). However, this is not likely in high-tech markets, where differentiation is used as a means of competition. For example, system usability could be a good way of differentiation in the cellular network markets, since the present systems are not very good in that sense. A more likely alternative on high-tech markets, is that there is not enough feedback from the markets to steer the product development (market feedback aspect). If the fast development is not steered by sufficient feedback, the products will flow down from the fitness hill.

In species with asexual reproduction, there are always some deleterious traits that are passed on and cannot be eradicated by evolution, if they are coupled with some useful traits. On the other hand, in species with sexual reproduction, like humans for example, recombination can combine the good traits of different individuals and thus accelerate exploring and finding the good solutions (one of the good sides of sex). In this regard, engineering seems to resemble more the sexual reproduction. Of course, in engineering there is no biological evolution (not yet anyway), but the engineers can, at least in theory, combine the good features without the bad, and in this way even find the global maximum of fitness. However, this becomes more and more difficult when the systems grow larger, since finding those good properties and avoiding the bad ones takes increasingly more effort. Consequently, this ideal combining cannot be done indefinitely, and there is an error catastrophe waiting around the corner.

By continuing with the natural model, we can say that mutations in nature are random. The probability of mutations increases when the number of traits in the organisms increase, since there are more genes, which can mutate. Due to the random nature of the mutations, they tend to push the adapting population to every direction (see Figure 16). If the population is on top of a hill and the mutation rate is high, the population begins to flow down, unless the counterforce (selection) pushes it back (see Christensen/Collier/Hooker 2000[172]). However, the increasing number of traits means increasing number of mutations, and when the number of traits (system size) exceeds a certain limit, selection is no longer able to push the population to a fitness peak and keep it there. The principle is also presented in Eigen/Schuster 1979[136].

The requirement is that at least one perfect copy (on average) must be made at each replication (Maynard Smith/Szathmáry 1999[137]). If there are n symbols, this means that the probability of an error when replicating must not be greater than $1/n$. For example, if the genome length increases from 1000 to 10000, the maximum mutation rate decreases from 1/1000 to 1/10000. If the mutation rate exceeds the limit, the population falls down – either smoothly or in a large jump - from the fitness hilltop and ends up on a level where the raising force (selection) and the sinking force (errors) are in equilibrium (see Figure 16). Certainly, it is entirely possible that the system was not even on a local maximum of the adaptive landscape in the first place, but it is definitely not climbing any higher, unless it started from below the equilibrium of the two forces. The population ends up on the same level regardless of whether it was ascending or descending.

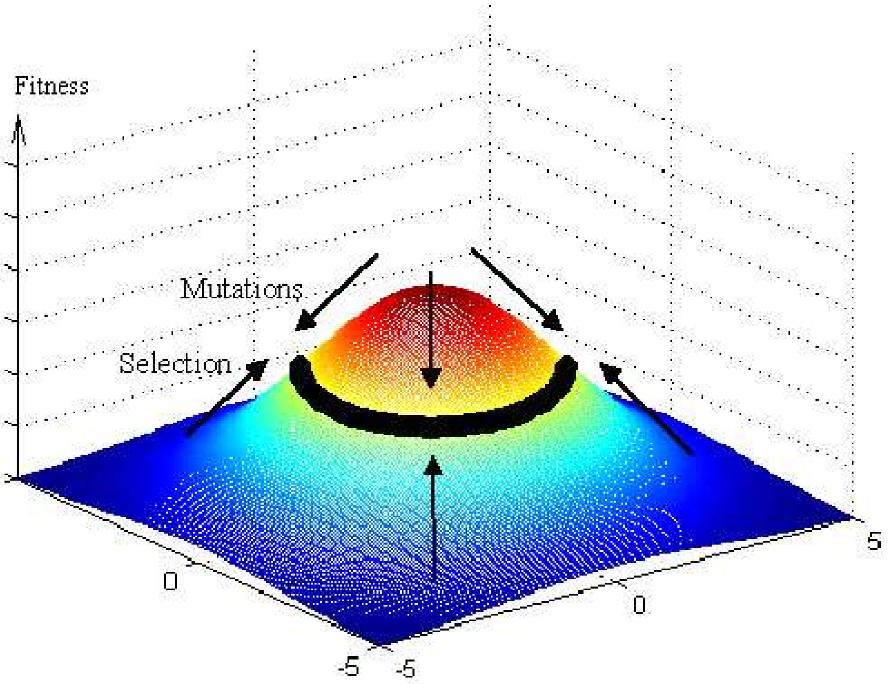


Figure 16. A simplified picture of error catastrophe in static landscape. Mutations happen randomly (omni directionally) and tend to drive the population away from the top. Selection picks the fittest and tends to drive the population uphill. If selection is too weak, the population flows downhill.

If we study this in the engineering/business domain, we can say that when a complex product reaches a certain size, the development of the system becomes too difficult increasing the risk of errors. This situation is further amplified by low feedback from markets. Consequently, the developed product can be badly matching with the market requirements and hence represent poor fitness in the landscape model. Singh 1997[138] showed that businesses developing technologies with high complexity had significantly higher risks of failure than businesses developing low-complexity technologies. This is understandable, considering that the increasing complexity must be accompanied by decreasing number of errors in order to maintain the level of fitness. Here, the term 'error' is to be taken literally and not as a synonym of 'modification'.

4.2.2 Complexity catastrophe

The second case of the catastrophes is the *complexity catastrophe* in which the selection can be strong enough to pull the adapting system to a fitness hilltop, but the hills fade away due to conflicting constraints. This situation occurs when there is a heavy cross-coupling between the traits (K is large) in the complex technical system. At first, the grave cross-coupling leads to a more rugged adaptive landscape with more peaks, as the complexity increases. Then, due to increasing number of conflicting design constraints, the peaks become ever-poorer compromises eventually forming mere small bumps (see Figure 17). As every trait depends on all the others, all solutions are compromises of all the cross-coupled traits.

As a simple example, we could say that if design specifications require creating a battery with the present level of technology, which is very light and possesses a very high capacity, the specifications are conflicting with each other. Consequently, the solution combining the mentioned traits would not represent a very high hill, where both traits would be maximised.

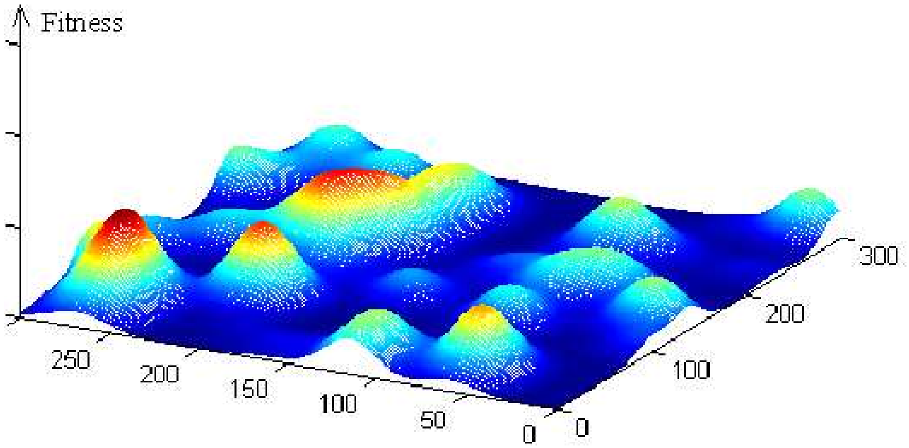


Figure 17. A simplified picture of complexity catastrophe. In the case of high cross-coupling of parts, the large amount of conflicting constraints eliminates ideal solutions and thus forces the hills very low. There is no longer a significant benefit to be on top of a hill.

This phenomenon has actually been evidenced by the technology research done by Fleming/Sorenson 1999[129] based on patent data (see Figure 18). According to the study, inventors can cope with increasing numbers of components as long as those components do not interact. By definition, the complexity is low in this case, since the K-factor is small. If we study Figure 18, we can see that the increasing number of components (N) first increases the multiplier effect (usefulness of the invention) very rapidly and then settles down to a very conservative improvement, whereas the increasing complexity (K) first enhances the multiplier effect and then starts to *decrease* it. As the K-factor becomes large, the complexity catastrophe sets in. Clearly it would be useful, if the increase of complexity could somehow be controlled when building complex technical systems. This also hints to the direction that we should find new ways of developing complex technical systems, for example growing some parts of them (see Shipman et al. 2000[131]) or other development principles found in nature.

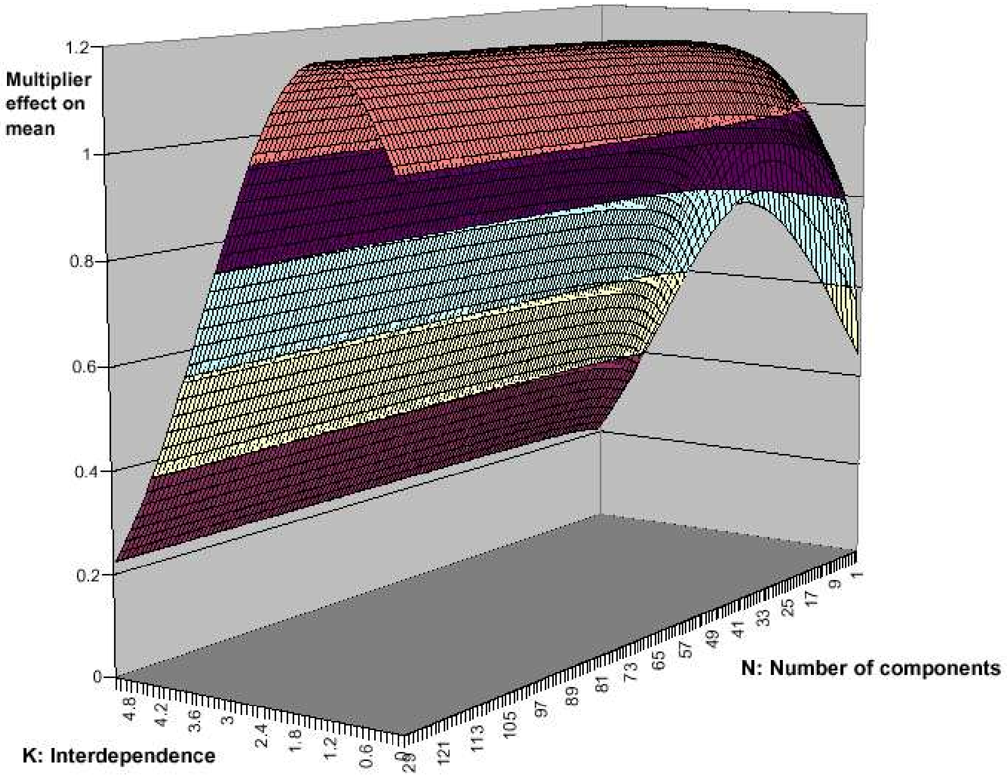


Figure 18. The multiplier effect and complexity catastrophe in technology. Increasing number of components (N) increases the multiplier effect monotonously, although very slowly after the initial rapid growth. Increasing interdependence of parts (K) first improves technology usefulness, but when further increased leads to a catastrophe. Source: Fleming/Sorenson 1999[129].

The complexity catastrophe is a serious problem (see McKelvey 1998[139]), where fast-cycled product development or good market feedback does not help. Kauffman, who has studied the complexity and emerging order extensively in biology, argues that there seems to be a construction requirement to make complex systems with many interacting parts, which remain perfectible by mutation and selection. The requirement is that each part should directly connect to rather few other parts (Kauffman 1992[126]).

4.3 The relation of complexity and system usability

We shall now extend the view to fitness and usability (refer to Figure 2 in chapter 2 'Definitions') by defining the relationship of complexity and system usability. As we can recall from the definitions, the conventional usability approach has been mainly to improve usability from a single user's point of view by creating good user interfaces. In system usability, we have a little more comprehensive approach. Here we concentrate on designing the whole system so that the operating organisation's effort for using the system is minimised (refer to the definition of system usability in chapter 2). This involves considering the complexity issues. The existing paradigm hints that in *system design* the user needs are considered only in connection with the user interface design (refer to Figure 3). In system usability, we consider the user all the time by managing the system complexity.

Complexity of the technical system itself and also complexity of the user interface have a great impact on system usability. If the inherent complexity of the system is not addressed when designing and constructing the system, we end up with a situation characterised by the peanut butter metaphor: The system is designed with no regard to complexity and everything is supposed to be handled by designing a user interface to create the usability. Even when the user interfaces are designed along with the system from the very beginning, they do not solve the problems created by ignoring the complexity issues.

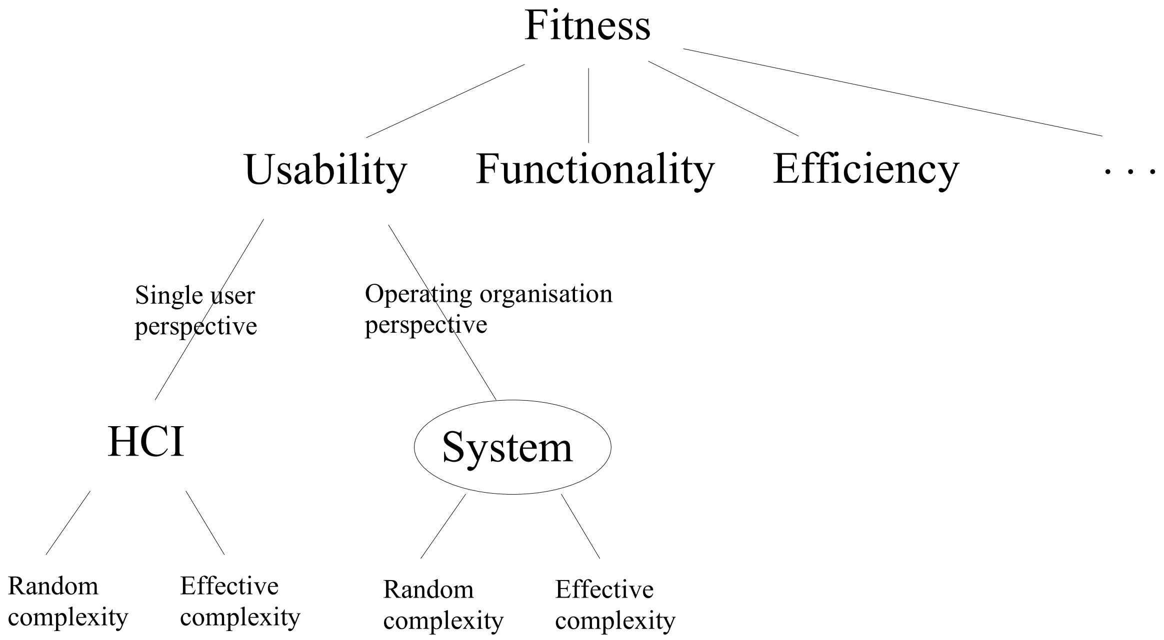


Figure 19. The relations of fitness, usability, system usability and user-observable complexity. The conventional usability paradigm concentrates on creating usability for single users with user interface approach. System usability has an operating organisation perspective to the whole system.

This issue can be elaborated by noting that when the inherent interdependency (K-factor) of the technology is high, the conflicting constraints reduce the fitness hills to mere bumps, thus radically reducing system usability. Even though we then climb up to the highest hill in the landscape by designing a superb user interface, there is no way we can reach superb system usability with the whole system. It simply is not available anymore. *We have eliminated the excellent solutions by letting the complexity (K-factor) of the system increase beyond the appropriate level.*

Also the user-observable complexity of the system may become very high, if the system is constructed without any concern of the complexity issues, and then trying to create the usability by the user-interface design. Designing a good user-interface for the very complex system might turn out to be very difficult and require a lot of effort. Another issue is the complexity of the user interface itself, which should also be considered. Both of these aspects should be considered in order to create good system usability in a complex technical system. In this research, we create guidelines for considering the complexity (see Table 5 in chapter 5.3 'Fundamentals of system usability' and also chapter 8 'Using the theoretical framework to find guidelines') to enhance the system usability of these systems.

4.4 Coevolution

Here we shall define the concept of coevolution, which also originally comes from biology. The concept is later applied to describing the coevolutionary relationship of companies developing complex technical systems, and also understanding *dynamic* adaptive landscapes. Coevolution also applies to many other levels of system development, for example, between various units within a single company or even on a personal level.

We can better understand the concept of coevolution by examining animal populations in nature. For example, if we take a closer look at a predator – prey relationship, we discover that the prey animals evolve to become e.g. faster and faster all the time, so that they can outrun the predators. On the other hand, the predators must also become faster (or develop other abilities) to catch the prey. In this way, coevolution enhances fitness of both the prey and the predator. It suffices for a prey animal to run slightly faster than other prey animals and for a predator to run faster than other predators. Usually, there are several different predator and prey species coevolving together and in this way affecting each others fitness landscapes.

In nature, organisms and species coexist in an ecosystem; each species has its own place or niche in the system. The environment contains a limited number and amount of resources, and the various species must compete for access to those resources. Through these interactions, species grow and change, each influencing the others' evolutionary development. This process of reciprocal adaptation is known as coevolution (Funes/Sklar/Juillé/Pollack 1998[140]). At its most basic, coevolution is defined as evolution in two or more evolutionary entities brought about by reciprocal selective effects between the entities (Ehrlich/Raven 1964[141]). In course material of Brown University (Anonymous[142]), coevolution is defined as follows: It is a change in the genetic composition of one species (or population) in response to a genetic change in another. More generally, the idea of some reciprocal evolutionary change in interacting species is a strict definition of coevolution. Brodie 1997b[143] defines coevolution so that it is reciprocally induced evolutionary change between two or more species. It is often analogised as an arms-race when antagonistic e.g. predator-prey, host-parasite, plant-herbivore. Gitzendanner's 2000[144] definition is quite similar to Brodie's: Reciprocal evolution of interacting species through natural selection. According to Eskikaya 1998[145], coevolution is evolution that involves successive changes in two or more interdependent species that affect their interactions. Hence, there is a continual evolutionary drive for better adaptation and counter adaptation, leading to an evolutionary arms-race between the species. As the Red Queen explained to Alice in Wonderland:

"...it takes all the running you can do, to keep in the same place. If you want to get somewhere else, you must run at least twice as fast as that!"

Nitecki 1983[146] defines coevolution the following way: Coevolution occurs when the direct or indirect interaction of two or more evolving units produces an evolutionary response in each. Baum/Singh 1994[147] talk about organisational coevolution mentioning also technological coevolution. In arms races or technological innovation, mutual competition can be coevolutionary, when competitors respond to and influence the actions of others. Each species has to evolve faster just to hold its own against others. In technology, this effect requires that continuous improvement be sustainable. It also means that in some situations it is advantageous to develop competitors, especially when legitimacy is being established (ibid.).

Murray 1999[148] describes coevolution between humans and technology. Humans are distinguished from all other living systems by their "tool-making" skills, progressively dominating the planetary surface through tool making, harnessing of stored energy and industrialisation. The latest phase in this coevolution of humans and technology is communications technology, which has progressively extended individual awareness beyond the face-to-face community since at least Gutenberg (ibid.). Beinhocker 1997[149] tells about economic coevolution and mentions technical innovations (such as the automobile) that produce ripple effects throughout the economy (oil industry, motels, suburbs etc.). McKelvey 1999[155] defines coevolution in business by using the biological definition of Roughgarden 1976[150], so that it means mutual causal changes between a firm and competitors, or other elements of its niche that may have adaptive significance.

As a conclusion, we can state that coevolution can be considered as a mutual adaptation of several agents (or groups of agents) living in close relationship. This relationship can be predator-prey – relationship or more like a symbiosis, the main thing is that the agents affect each other. The adaptation can be understood so that the agents live in their own adaptive landscapes, which are connected together via the coevolutionary links (see Figure 20). Through these links, the moves toward better fitness by any partaking agent in their own landscape will change the landscapes of the others thus causing them "development pressure".

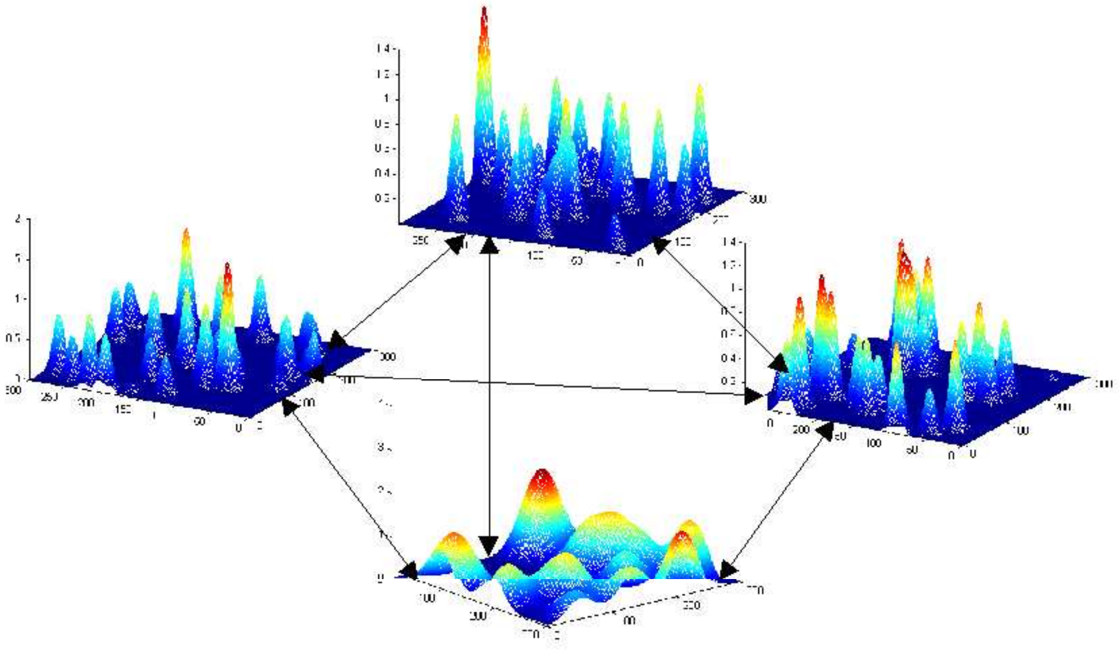


Figure 20. The adaptive landscapes of four companies maximally linked together with coevolutionary links (K = N-1). When the companies strive for better fitness (higher ground) in their own landscapes, these actions constantly mould the landscapes of the others via the coevolutionary links.

4.4.1 Nash equilibrium

Kauffman made simulations of coevolving metasystems assuming that each system acts in turn, while keeping the other systems constant for a while (Kauffman 1992[126]). Such a coevolutionary process creates two behaviours. Either a) the systems keep coevolving in turns, or b) the coupled metasystem attains a steady state, a so called *Nash equilibrium*, where the local optimum of each system is consistent with the local optima of the other systems via C couplings. Actually, Kauffman uses the term 'analogue of a pure-strategy Nash equilibrium', because he assumes that each system is constrained so that, at each moment, there is only some of the traits accessible by mutation. In a true Nash equilibrium, each agent can choose any one of its possible actions at any time. The concept of *evolutionary stable strategies* (ESS) is a further refinement of the Nash equilibrium. In this concept the systems stop changing at a balance where any other combination of traits tried by any system would be less fit. In other words, it is a stable, optimum combination of traits for the coevolving systems (ibid.). This is not very likely in practical complex technical systems, since the dynamic landscape changes before all the accessible combinations have been tried.

In Kauffman's simulations (Kauffman 1992[126]) reaching the Nash equilibrium took hundreds or thousands of generations. Similar results can also be found in Ray's article (Ray 1994[151]). Studying fitness at the Nash-equilibrium is not practical in fast pace hi-tech business, since there is not enough time to reach the equilibrium before the landscapes change again. We shall concentrate on the dynamic landscape in the next chapter.

4.5 Dynamic adaptive landscapes

Now, we shall take a closer look at the fitness landscapes in a dynamic situation, where the coevolution between the various adapting entities continuously moulds their adaptive landscapes making it necessary for all the players to constantly track the changing landscape in order to keep the products at high level of fitness. This tracking is not always successful and causes catastrophes. By realising the existence and the way how these catastrophes emerge, makes it possible to avoid them.

In a general case, where the variables (complexity of the system, dynamics of the landscape and mutation rate of the system) get some values depending on the technological/business situation, the required selective force (market feedback) must be sufficient to balance the effects. There are many different combinations, which the company must be able to handle. For example, if the mutation rate of the product development is too low, the company is not able to keep up with the dynamic landscape. If, on the other hand, the mutation rate is too high, the risk is to fall down from the hills due to insufficient selective force (refer to chapter 4.2.1 'Error catastrophe').

According to Nilsson and Snoad (Nilsson/Snoad 2000[152]), there are four variables affecting the survival of a population in a dynamic adaptive landscape, where coevolution is involved: 1) Landscape changing speed, 2) Complexity of the adapting entity, 3) Mutation rate and 4) Selection power. In technology, the mutation rate can be considered as the modification rate and the selection as a market feedback. The characteristics of the error catastrophe in dynamic conditions can be seen in Figure 21. The figure has been created with certain assumptions, which are not important in giving a qualitative view.

The error catastrophe occurs 1) When the rate of modifications (mutations) *increases* beyond a certain level, and the selection is not strong enough to keep the product on the fitness hill. 2) When the rate of changes *decreases* below another level, and the product can no longer track the changes of the dynamic landscape. In this case, the hills move away from under the entity (complex technical product) and hence the product fitness decreases. Here, we have to note that the latter case is only possible if the landscape is dynamic. In the figure, the selection power has been kept constant. Of course, the limits of the catastrophe can be driven further by increasing the selection power (market feedback).

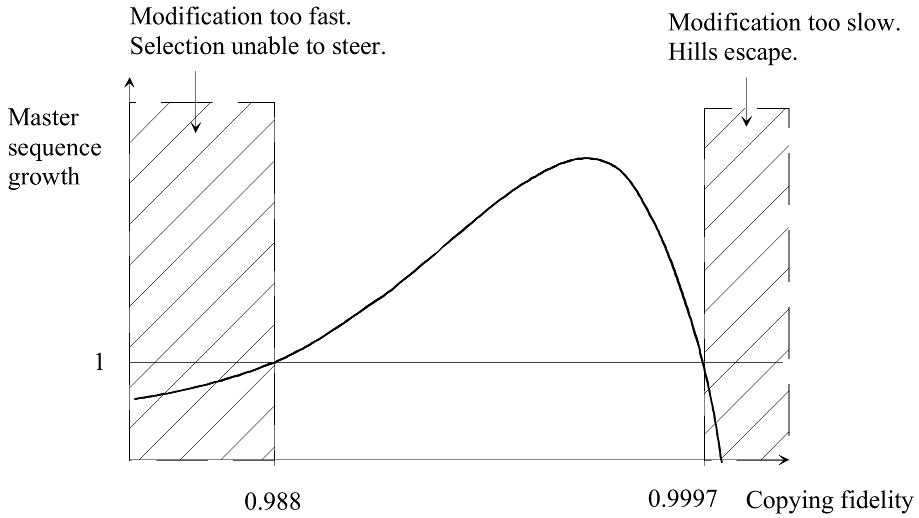


Figure 21. Below the lower limit (0.988), changes are generated faster than the feedback can steer => products flow downhill on the fitness hill. Above the higher limit (0.9997), the products cannot track the dynamic landscape fast enough => hills move out from the product. The figure is from Nilsson/Snoad 1999[152].

We can apply the analogy to technology/business to gain some explaining power to the situation. *We could say that*

a) when a product has a certain complexity and

b) the company has a certain product development rate and

c) the dynamic landscape is changing at a certain speed, *there is a minimum selection power (market feedback) required to track the changes.* The higher the product complexity or the faster the product development rate or the faster the landscape is changing, the stronger is the required market feedback (selection power).

We could also say that the purpose of the product development of a hi-tech company is to track the changes in the fitness landscape and try to keep the products on the highest fitness peaks. In order to achieve this, the product must have a change rate appropriate to the regularity and rate of change of its fitness landscape (Christensen/Collier/Hooker 2000[172]). This is not easy, because presently there is no way of measuring the landscape and due to the difficulties in adjusting the parameters, such as K (for a study about the effects of K in technology, see Fleming/Sorenson 1999[129]). Of all the mentioned variables we cannot completely control each and every one. Let us study the variables in the ascending order of control possibilities.

The changing speed of the landscape is beyond the control of any single player, since it is a property of the metasystem to which all the players contribute. There is a little more control over the (effective) complexity of the product (technical system), but even that usually must, more or less, match the complexity of the competing systems. The mutation rate (product development rate) can be controlled to a higher extent, but also that is not totally free of limitations. For example, the product development cycles must be fast enough to match the business in question or else the company is not completely set to compete on the markets. This leaves us with only one parameter, which can be (almost) freely changed: The selection power consisting of market feedback. This should be frequent and abundant to secure enough selection power for the product development. In complex hi-tech products, it would be very valuable to get the customer feedback already very early in the development, even before any product design has been started. This would involve, for example, testing of alternative product designs via the Internet.

A good example of very efficient use of selective power is those hi-tech companies who let the customers do β-testing of their SW-products via the Internet. In this case, the customers actively *participate* in the product development. There are also other ways of increasing the selective power, but they all involve utilising the customer feedback within the development cycle. Some day it might be possible to create a virtual experience of using the product to the customer, but today perhaps animated, audio-visual multimedia descriptions of the product features could be used to provoke some feedback in the product development phase.

4.6 Coevolving systems

Here, we study coevolution in case, where several supplier and operator companies coevolve together, constantly moulding each others fitness landscapes. There are also other players in minor roles in addition to these lead role players, such as standardisation institutes, legislators, emerging new technologies etc. In this dynamic environment, it is difficult to stay at very high level of fitness, since the landscape may suddenly change and wipe the adapting system down from its position. This is particularly true, when some disruptive technology emerges causing a lot of havoc. In such a situation, entire industries can fall down to a valley of fitness landscape and face extinction (see Schumpeter 1934[153], Anderson/Tushman 1990[154]). This is characterised by the concept of *punctuated equilibrium* (e.g. Kauffman 1992[126], Beinhocker 1997[149]), meaning that the times of relative calm and stability are interrupted by stormy restructuring periods, or punctuation points.

In a coevolutionary relationship, the fitness landscape of each system is affected by the other systems. This also means that the fitness of the neighbouring entity indirectly depends on the fitness changes of all players in the metasystem. If we use the NK model to study the phenomenon, we can connect the landscapes of S systems into a larger metasystem. Each trait in the individual systems depends on K traits internally and on C other traits in each of the remaining S-1 other systems. This is demonstrated in the simplified example in Figure 22, where we have a metasystem of two system suppliers and one operator. Typically though, there are many more players living in coevolution within the metasystem.

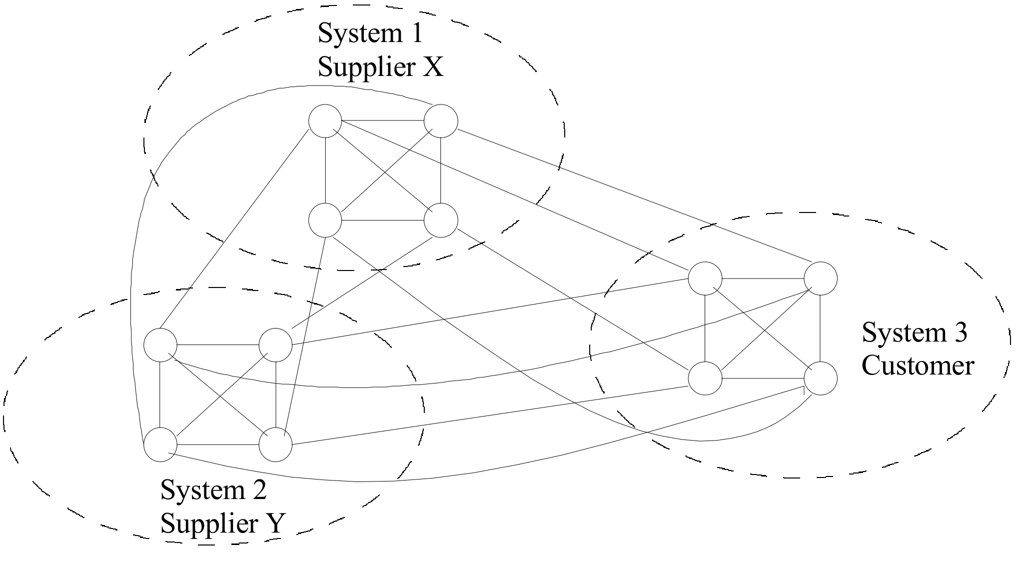


Figure 22. Coevolution of three systems. N=4 (four traits), K=N-1 (maximally coupled internally), C=1 (each trait is connected to one trait externally) and S=3 (three systems). The three systems deform each others' landscape via the connections between the systems.

The coevolutionary NK model fits well the business world. For example, McKelvey 1999[155] argues that the assumptions of the model are actually more straightforward for firms than for organisms. We shall now briefly study coevolution in the hi-tech environment.

Coevolutionary relationships occur, for example, when complex technical systems are created and marketed to the operators. This relationship is a *supplier-user* relationship. The product development of the supplier company creates new complex technical systems with certain traits. The manufactured systems are offered on the markets, which will evaluate them and either accept or reject them (see Figure 23). Fitness of the operators is affected by the capabilities of the product, when they use it in their own business. The operators judge the products based on the success in the operating tasks, and give feedback (selective force) to the suppliers. As a consequence, this feedback acts as a selective force shaping the course of product development into a certain direction. We could say that this information is a precious asset, which should be exploited to the maximum in the company, and not left partly unused, e.g. due to the inflexible development process. The acts of the operators also affect the dynamic landscape of the suppliers via the links.

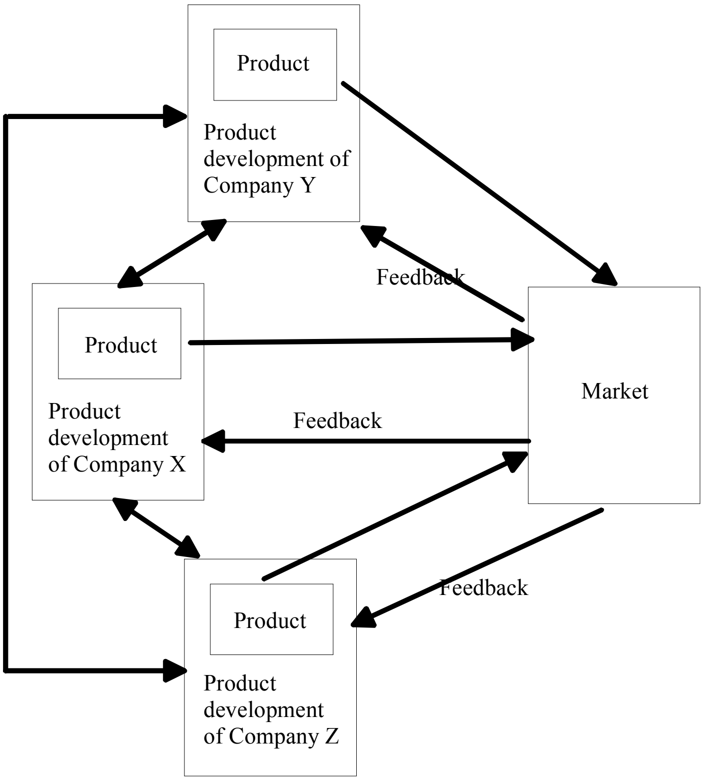


Figure 23. A simplified drawing about coevolving systems in business. Any adaptive move by one of the systems affects the others via the coevolutionary links and deforms their adaptive landscapes.

In addition to the supplier-user relationship, there are also two *competitor* relationships among the major players: *Supplier-supplier* and *operator-operator* relationship. Suppliers compete on the technical product markets by creating better and better systems with higher technical fitness. The acts of the competitors shape the fitness landscapes of each other via the coevolutionary links. In a similar fashion, the operators compete in the operating business by using the equipment they have purchased, and thus mould each others landscapes. Fitness of the technical systems can have a major effect on operator business. If the system usability of some supplier's technology is very good, the operating effort is decreased, which in turn reduces operating expenses and increases competitiveness in the operator world.

In natural systems, if two species are in direct competition and one of them is constantly better than the other one, this arrangement usually means extinction for the loser. In business, however, the situation is not always quite like that, since the players on the markets are not necessarily behaving completely rationally and there can be other factors, such as marketing, which enhance the surviving chances of inferior products. It may not even be desirable to drive competitors to extinction. By accommodating Kauffman's biological theory (Kauffman 1995[134]), we could state the following. When a company is driven extinct, the event may trigger an extinction avalanche. The removed company is replaced by an invader, which is new to the niche and typically not at a local peak, and therefore adapts in new ways. These moves change the fitness landscapes of the other players, typically lowering their fitness. As their fitness is lowered, they become more vulnerable to successful invasion and extinction.

In coevolution, also the number of coevolving systems seems to be a significant parameter. Kauffman's simulations show that, as the number (S) of mutually coupled systems increases, the mean fitness falls and fluctuations to very low fitness increase. Therefore, if S is too large, the coupled metasystem will fluctuate dramatically and lead to the extinction of some of the systems, which in turn lowers S. Then, as S is lowered, the metasystem behaves less chaotically and mean fitness of all systems improves. This can be considered as an in-built control loop, which automatically keeps the number of interconnected systems within a certain range. The same phenomenon also happens in industry/business (McKelvey 1999[155]). If there are too many players in the metasystem, it becomes unstable and the shakeouts will lead to extinction of some of the players. After this, the winners are doing fine again. Thus, the most stable and high performing coevolutionary pockets tend to be those having relatively few firms (ibid.). According to Kauffman 1992[126], each player maintains roughly a constant number of connections (C) to other players, regardless of the number of players in the web. Observations in nature suggest that each system (population) remains coupled to three or four other systems (S=4, 5) (ibid.).

When applied to coevolution between firms (McKelvey 1999[155]), Kauffman's simulations suggest that a) Increasing K is not good, unless the opponent has a high K or fosters a high C, b) If the opponent raises its K or C, the Nash equilibrium occurs faster and it is better to have low K (low K means higher fitness peaks), c) If the opponent does not raise K or C, the equilibrium does not occur quickly and a low K firm will lose its advantage. In general, it seems that keeping one's internal and external coevolutionary interdependencies just below that of the opponents is the best strategy (ibid.).

As we have noted, in a coevolutionary relationship of companies, an adaptive move by one player to reach higher fitness projects onto the adaptive landscape of the other players and alters those landscapes. This means that the product development of any company can have a radical effect on the fitness landscape of other players' products, if they come up with something that has drastically better fitness (more readily accepted on the markets). This superiority will be noticed on the market, which is the ultimate judge for fitness, and the innovating company will get a temporary advantage. Consequently, the other players improve fitness of their products to match the new product. This leads to a situation called the Red Queen or rat race, wherein the coevolving products keep relative positions while their properties evolve continuously. Coevolution also benefits all the participating players (Kauffman 1992[126]), and hence keeping up strong coevolutionary links (C) with the major players is advantageous. On the other hand, too many coevolutionary links (K, C) increases the interdependence and involves the risk of a complexity catastrophe, particularly if the other players have lower K and C.

5. CONCLUSION OF THE THEORETICAL FRAMEWORK

5.1 Basic concepts

The common denominator in the theoretical framework is *complexity*, which means multiple interrelations between many agents. Complexity is involved on many different levels, since complex systems are hierarchically structured. Each agent is complex, the systems consisting of agents are complex and the metasystem consisting of the systems is complex. Complexity creates emergent phenomena, which are virtually impossible to predict based on the understanding of components. Complex adaptive systems also feature rather high level of autonomy in the subunits of each layer.

The system usability point of view in this framework is connected with complexity, since complexity must be understood by the technology suppliers in order to create usable complex technical systems (see Figure 24). Unless complexity is efficiently managed, it is not possible to reach good system usability, and in the worst case the excessive complexity leads to a complexity catastrophe.

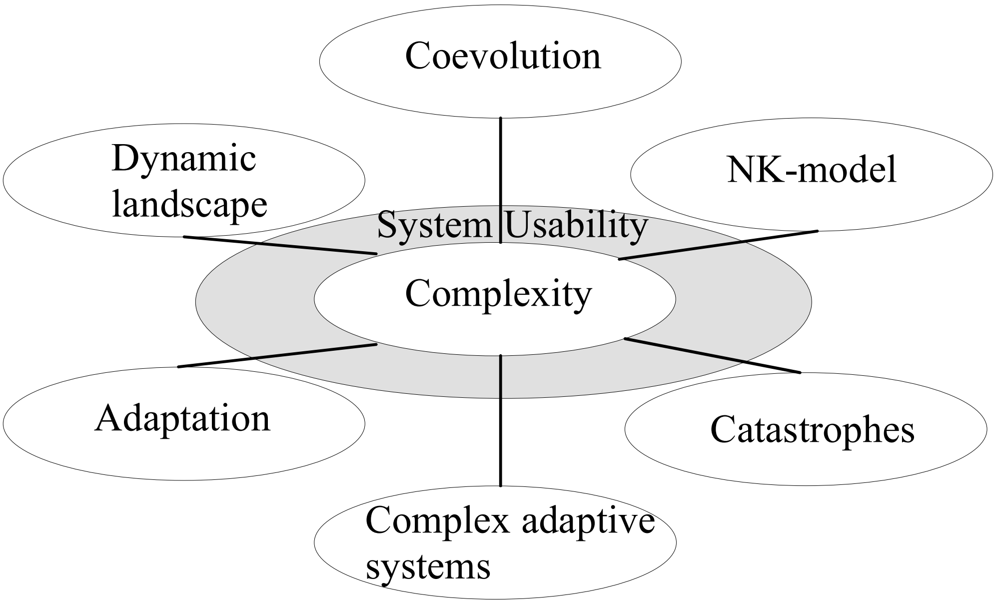


Figure 24. The main concepts of the theoretical framework. The focal concept is complexity, which must be understood in order to build usable complex technical systems.

Adaptive landscapes, which are fitness landscapes of the agents, are dynamic since the agents live in *coevolution* with each other. The multiple links between the agents affect the landscapes of the other agents causing constant need for *adaptation*. A natural agent living in coevolution with other agents is a *complex adaptive system*, which is able to cope with the dynamic situation. In nature, the agents usually live in populations. Sometimes the adaptation is not successful and leads to a *catastrophe,* when a whole population may vanish. There are basically two types of catastrophes: *error catastrophe* and *complexity catastrophe*. An error catastrophe occurs when the rate of mutations (modifications) overcomes selection (feedback). When error catastrophes occur, the population is not able to stay on a fitness hill and falls down to a level of poor fitness. A complexity catastrophe occurs when the number of interdependencies between the components becomes high. When a complexity catastrophe occurs, the fitness hills get very low and there is no longer a benefit in being on a hill. The *NK model* is a means to study complex systems by using two variables, N (number of parts) and K (cross-coupling of parts). By changing the variables of the model, we can create different fitness landscapes with various degrees of ruggedness to simulate different situations or model existing systems.

We also introduced the concept of *user-observable complexity* to describe how the inherent complexity of the system is observed by the various users. Complex technical systems are seen very differently by different user groups, and user-observable complexity is a subjective experience of the system complexity at the level of access. User-observable complexity can be divided into *random* and *effective* (non-random) complexity. The effective part characterises the potential capabilities of the system, whereas the random complexity only makes things more difficult without any useful contribution. As the user-observable complexity tends to increase, it would be useful to keep it in control. Reduction of user-observable complexity can be done, for example, by eliminating random complexity and also by compressing effective complexity. *Metasystem transitions* have the capability to isolate the users from lower levels having high number of details, and in this way reduce user-observable complexity. Technical realisations of *complex adaptive systems* have a lot of potential in realising metasystem transitions and improving system usability in complex technical systems, since they allow elimination of the low level manual operations and enable high level of subunit autonomy.

System usability is a comprehensive approach to understand usability of complex technical systems via system complexity. System usability is defined based on the total effort of all user groups required for operating the system.

5.2 Applying the concepts to technical product development

The companies producing complex technical products live in their own adaptive fitness landscapes searching for product improvements to get higher in their own landscape. The companies (suppliers and customers + some other players) are tied together with coevolutionary links, and hence an adaptive move by some player changes the landscapes of the other players via the coevolutionary links. Consequently, the landscapes are dynamic making it necessary for the adapting entities to constantly track the fitness hills when pursuing better fitness. As complex systems are combinations of very many potential parts, the total solution space is astronomical. This also means that the global fitness maximum is impossible to find. However, there are many excellent combinations, which the technology companies are searching for. In a sense, the product development is constant hill climbing in a dynamic, ever-changing landscape, where the moves are made by selecting different feature combinations for the products from a vast solution space. The pursuit is not always successful, and certain conditions lead to error and complexity catastrophes. In such a case, the survival probability of the product or even company is drastically reduced. These catastrophes can be avoided by applying certain rules. The prime directive is that the K-factor in any complex system should be kept relatively low. In other words, any part of a complex system should be directly coupled to relatively few other parts.

In complex technical system usability, it would be useful to complement the HCI-approach by complexity understanding. In chapter 8, there are some examples of applying the theoretical framework to complex technical product development. However, since designing technical applications is not the purpose of this research, the examples are not yet taken to the engineering level.

5.3 Fundamentals of system usability

As we have noted before, developing complex technical systems involves the risk of catastrophes. In a dynamic landscape, there are two possible *error* catastrophes. If the development rate is high, there is a risk to fall down from the fitness hill due to insufficient selective force. If the development rate is slow, there is a risk that the fitness hills escape leaving the product again to a state of low fitness. There is also a risk of *complexity* catastrophe, if the interdependence of the technologies or parts is too high, thus causing the fitness hills to diminish. In this chapter, we list some means to avoid the catastrophes and pave the way for better system usability.

The feedback from the markets should be abundant and frequent, since this is the way to improve selection (see chapter 4.2.1 'Error catastrophe') and thus guide the product development. For example, virtual models on the Internet can be used to provoke feedback for products that do not exist yet. Even these measures are not always sufficient, and then the only way is to send probes (test products) to the markets and get the feedback in this way. While doing this, one must remember not to make too expensive experiments.

In addition to market feedback, enhancing the links between the different players in the metasystem is useful, since coevolution improves fitness of all players. This would mean strong links both internally within the company and also externally, for example, between suppliers and operators.

The dynamics of the product development process should match the dynamics of the business. In high-tech business, the dynamic landscape changes fast, and if the time constant of product development is much longer than that of the adaptive landscape, the company is not able to track the changes in the landscape. Hi-tech companies should use development processes that allow rapid adaptation to the customer feedback.

Only a small number of parts should be directly connected to each other to keep interdependence (K-factor) from rising too high (see chapter 4.2.2 'Complexity catastrophe'). This can be achieved by using modular technology, in which each function is manifested in its own module having relatively few connections between the modules. It is also wise to create a hierarchical structure, where the complexity is handled locally by using autonomous subunits and not concentrating all the control to the highest level.

If we take a lesson from nature, we identify and remove the possible random features (as opposed to non-random features or effective complexity) of the system and in this way optimise it. In addition, effective complexity can be reduced by innovating simpler ways of achieving the same and even higher goals. These innovations should not be discarded simply for not offering any new functionality.

Presently, the technical systems are rigidly constructed before taking the systems into use to contain all possible alternatives for use. Thus a large number of parameters must be tuned when running the system in order to adapt it for the various conditions. Particularly in dynamic environments, this is difficult and laborious. Adaptive technology has the capability of run-time learning, which enables the system to adapt to the dynamic situation without manual parameter optimisation.

We can also use the concept of consecutive metasystem transitions (Turchin 1977[100], Heylighen 1994[98], Heylighen 1999[99]), meaning that whenever the user-observable complexity reaches a certain limit, we introduce another higher level of control and hide the lower layers by architecture (e.g. by autonomous subunits). This will keep the user-observable complexity at the metalevel relatively low, even though the effective complexity is increasing. We can understand how difficult the situation would become without these transitions, if we consider e.g. vital bodily functions of human beings. For example, if the working of the immune system, body temperature control or breathing were not autonomous functions but constantly controlled at will, life would become extremely difficult if not impossible.

Table 5. Fundamentals for system usability of complex technical systems

Improving feedback from markets to enhance selection
Enhancing coevolutionary links between the players
Matching product development process with the dynamics of the business
Reducing cross-coupling (K-factor) by modular technology
Using hierarchical structure with autonomous levels
Removing random complexity
Reducing effective complexity by innovations
Using complex adaptive system technology
Using metasystem transitions to manage user-observable complexity
System understanding

5.4 Meaning of the theoretical framework

The present usability paradigm understands usability of technical systems more or less through human computer interaction and user interfaces. In this theoretical framework, we have studied a new approach for understanding usability through system complexity and the total organisational effort required for operating the system. We have also found some guidelines that could be used for complex technical system development.

As the technical systems are getting more and more complex all the time, it is not enough to understand only technical functionality, but we also have to understand the essential characteristics of complexity. This is useful, since it provides possibilities for managing the complexity and its implications. For example, the interdependence of various parts or technologies in the system should not be excessive in order to avoid the complexity catastrophe, or the user-observable complexity should be kept sufficiently low by controlling its two components. If the complexity of the system is not considered, it is possible that the fitness hills are reduced in the system design by excessive interdependence of components/features and no user interface design can correct the situation anymore. In such a case, the users would have to invest a lot of effort for achieving their goals, since the efficiency of the system is not as high as it could have been with the complexity issues considered. This happens even if the new user interfaces allow easy and straightforward usage of the created (but not effective) functionality.

Nature has been developing complex systems for a very long time. Although we do not have a complete understanding of how everything happens in nature, we know many principles used by nature when building complex systems, such as optimisation, innovation, modular design, tinkering, spontaneous order, coevolution, complex adaptive systems, etc. Applying the observed principles would allow us to more successfully create increasingly complex technical systems.

6. EMPIRICAL RESEARCH

6.1 Introduction

The purpose of the empirical research was to validate the theoretical framework in complex technical system industry. The scope in the research is hi-tech products, as the framework is designed for dynamical situations. The commodity business does not involve continuous development of new innovative products and hence the dynamics of the field are slower. Cellular telephone networks was selected for the studied dynamic hi-tech industry, since the researcher had a good pre-understanding of the technology and operators. However, since the network operators were not willing to tell about the equipment of all suppliers, we had to settle for the technology of one supplier only. This means that the multivendor environment, in which every operator functions in practice, is not very visible in this work. The scope of the research was also limited to system operating in the operating organisation and product development in the supplier-operator relationship and hence, for example, the standardisation phase is not studied.

The empirical research is built on the basis laid down by the theoretical framework. The aim is to get an idea of the complexity involved in network operating, chart the present system usability of cellular networks from operators' point of view, and also learn more about the coevolutionary nature of technology development. In addition, improving the system usability by finding process and technology improvements was to be studied. The data were gathered from four different operators so that the observations were done first with one operator and then the knowledge was complemented by interviews with the other three. The field work was done by studying GSM network operations and it was limited to BSS and OSS subsystems (see Figure 25 and Figure 26). In other words, NSS was left out of the scope.

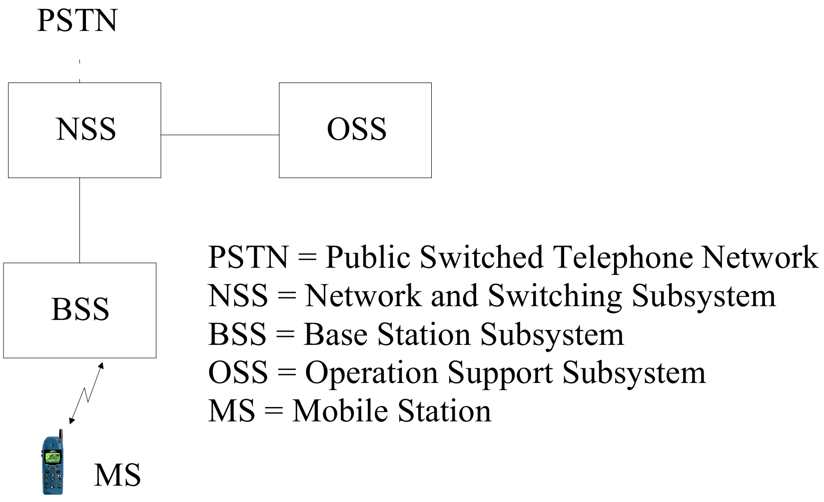


Figure 25. A simplified view of GSM on subsystem level.

The data were gathered as a combination of interviews with many different working groups and observations in the network control room of one operator (for a view to a control room see Figure 32). Due to the rather central role of the network management personnel in network operations, the observations were done by observing the NMS[a] users in the control room. This gives a good view to the everyday operations, as this group is in frequent contact with almost all other groups. We also cover the work of the network management group and the management system itself a little more extensively in this description.

Many things are involved in the study. Most important are the systems the operators are using to monitor and maintain their networks, the operating processes, the co-operation with the various groups in the operator's organisation and the role of the control room in the operation. In spite of the rather central role of the NMS, it is good to remember that the actual telecommunication system conveying the traffic consists of the BSS and NSS. The network management system is there to enable the maintaining of the telecommunication.

Modern cellular networks are very complex systems (see Bendett/Neelakanta 2000[156], Neelakanta/ Deecharoenkul 2000[157]). The need for making the work easier for the planners has been recognised by major operators. For example, British Telecom tells (Shipman/Shackleton/Harvey 2000[131]): "In order to meet the increasing demand on BT's telecommunications networks, they must be *grown*[*]. The ever-increasing complexity of this task [network planning] necessitates tools to aid the network designers and planners". Particularly in BSS, there are many complex phenomena (see chapter 6.4 'Simulations').

Some part of the radio network complexity is not directly visible to the network management group, as they do not have to understand the BSC[b]-algorithms and all the complicated and inter-related parameters of the BSS when performing their tasks. They can only see some *implications* of this complexity, for example, bad quality of service when the network is not performing well. The radio network planners are directly coping with the radio network complexity, when optimising the network and tuning the parameters. It is a major challenge to comprehend all the radio wave propagation issues in different environments and conditions, the control algorithms and the inter-related parameters of the supplier's system, as well as all the combined and co-working features in an multi-layer/multi-frequency/multi-technology system. In addition, this situation gets more

---

[a] Network Management System, see chapter 'Network management system'.

[*] Italics added.

[b] Base Station Controller.

complex all the time, as the new features are introduced in the new releases and also new, more complex technology generations are created.

6.2 Cellular networks

Generally, a cellular network, such as GSM, consists of three subsystems: NSS, BSS and OSS (see Figure 25). In this research, the system was restricted to BSS + OSS, since the researcher had to limit the scope. In Figure 26, we can see the contents of BSS and OSS at the level, which is sufficient for understanding the empirical research.

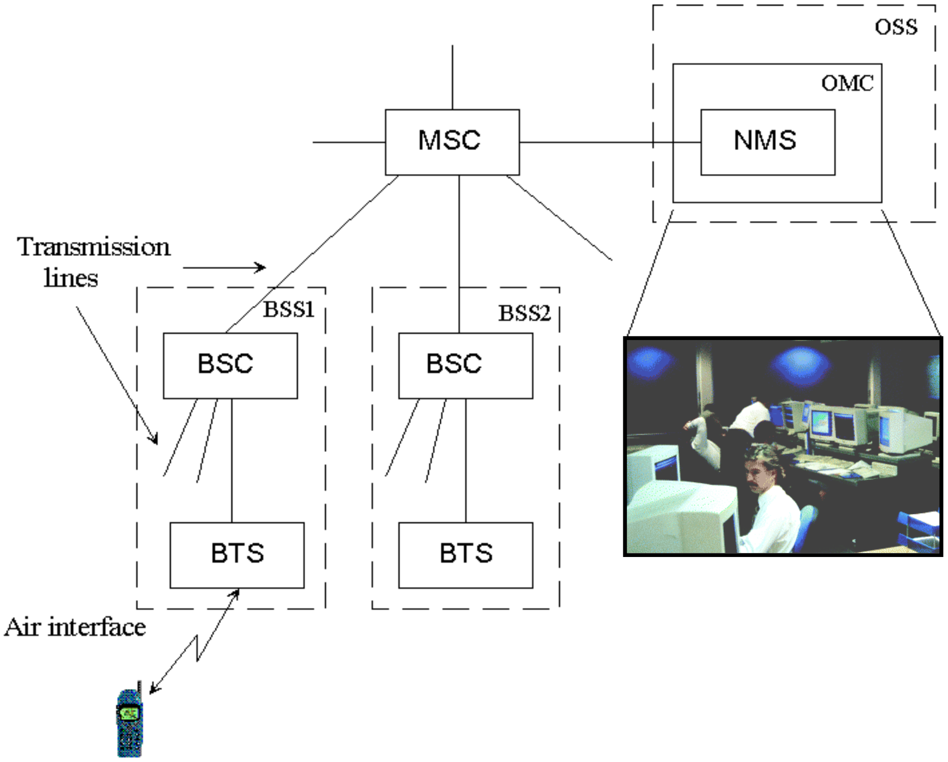


Figure 26. Second detail level of BSS and OSS. Refer to Figure 25.

If we compare a mobile telephone network to a fixed telephone network, we find that some additional parts are necessary in the network. This is because the telephones are *mobile* phones instead of fixed ones, and the system must provide seamless service regardless of where the users happen to be travelling. For one, the interface between the user and the network is realised as a radio interface to allow the movements. This is done with the aid of radio base stations, which in the GSM system are called Base Transceiver Stations (BTS). Each of the base stations covers a certain geographical area, which is called a cell (hence the name cellular network).

There must also be some kind of *mobility management* to track the movements of the users and make sure that the users have uninterrupted service within the coverage area of the network. The tracking is done with the resolution of a location area, which typically consists of a few base stations, and is stored in the databases of the system. This tracking procedure is called location updating.

Radio resource management makes sure that the user always has an interference-free connection and that the connection is maintained when the user moves in the network. This functionality contains e.g. a handover and power control algorithm, which in the GSM system is manifested in the Base Station Controller (BSC). The algorithm takes care of switching the call to another base station when the current connection becomes too poor or a better alternative is found, as well as adjusting the transmit power of the telephones and base stations when needed.

The mentioned algorithm and other features in the radio resource management are a major contributor to the complexity of these systems, in spite of which they have been left out of any usability design. This is probably due to the present user interface oriented usability thinking. In addition, the base stations have been conveying all the low level fault messages without any analysis to the network control room, where the users have to deal with thousands of these cryptic messages every day. However, this latter problem has already started to improve, as the suppliers have designed the new base stations to have more intelligence than the earlier generations.

A network management system (NMS) is used to access the different network elements in the network (BSS and NSS). Many times these "network" management systems can only be used for element management (see Figure 27) and one vendor's equipment. The management systems are placed in the operation and maintenance centre (OMC), which is a centralised control place for various network management tasks.

In addition to the NMS, the operators still use a direct control to BSS (and NSS) via command-based Man-Machine-Language (MML). Although, the NMS, being equipped with a graphical user interface, is potentially much more efficient than the relatively low level MML, the potential is not fully realised. For example, the NMS applications are mostly not designed to accomplish any operator tasks, but only offer some functionality required in the tasks. Consequently, the operators prefer to design their own tools for some of the tasks. It is also obvious that the graphical tools are not handling large enough task entities without a lot of manual intervention. This is also a reason why the operators prefer to use their own tools for many tasks.

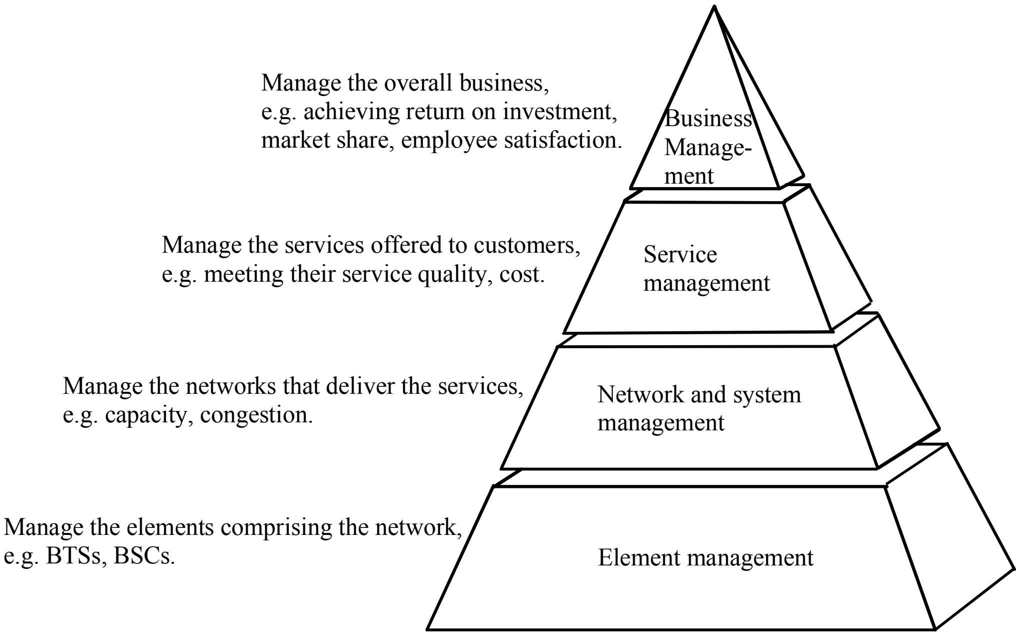


Figure 27. The four layers of management according to Telecommunications Management Network (TMN) framework (Adams/Willets 1996[3]). Hi-Tech Inc.'s NMS fits primarily in the Element Management layer.

6.3 Cellular network product development

In this sub chapter, we shall briefly describe what kind of product development process is used in the example company to develop the cellular radio networks. Hi-Tech Inc. uses currently the so called Stage-Gate product development model (Cooper 1990[158], see Figure 28), although some experiments have been made to modify the process. Stage-Gate models emerged from research in industries, where customer preferences and underlying technologies changed relatively slowly (MacCormack 2000[159]) (e.g. packaged goods, automotive development). This is an efficient process in an environment where new information is unlikely to emerge once the concept design has been frozen. In contrast, flexible models have emerged in industries, where new information on customer needs and underlying technology arrives continuously during development. The following description of the way of development is based on discussions in the work community of Hi-Tech Inc.

The present product development cycle starts, when some engineers from the product line of the supplier company visit the customers and collect the requirements in discussions with the engineers of the operator. Next, the requirements and suggestions are evaluated to find suitable feature candidates for the coming releases. Then, certain features to be implemented are selected and the rest (about 90 % of them) are either discarded or postponed for more distant releases. When the coming features have been selected, the 2-year development cycle starts, during which, no feedback is used and further modifications are not possible before the product is released and the cycle starts again.

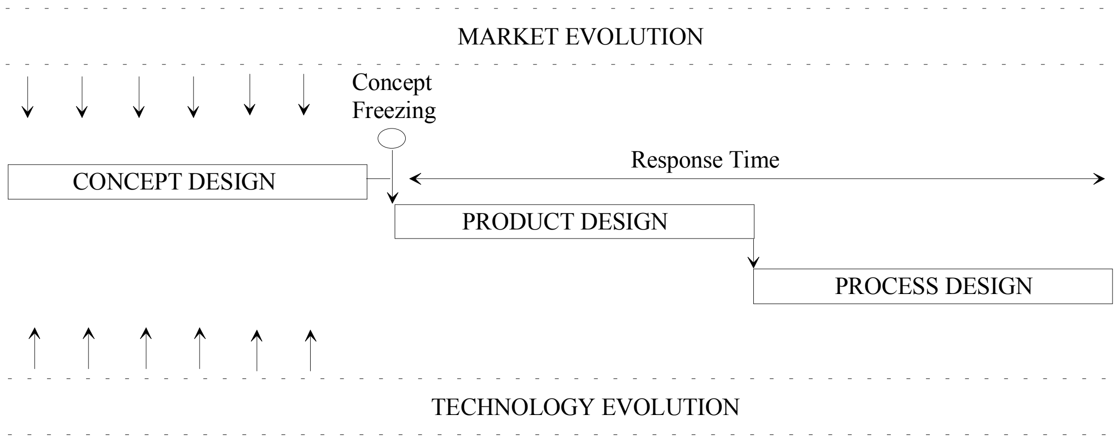


Figure 28. The Stage-Gate product development process (Cooper 1990[158]). Cost and time of late changes are high – late changes are avoided.

6.4 Simulations

In this subchapter, we shall present some simulation results to show that there are interdependencies between the network parameters. This is, by definition, complexity, and hence we can state that the system is complex. As discussed earlier, some literature references argue that cellular networks are complex technical systems. Since we had access to a GSM radio network simulator containing the same control algorithm as the real networks, we decided to search for interdependencies between the radio network parameters. This kind of dependency has not been shown by an experiment, although many experts have anticipated the phenomenon based on their experience in the field.

Since there are hundreds of parameters, it is not possible to check every parameter against every other. This is the basic problem with complex systems – the solution space is so huge that all combinations can never be tested. Therefore, we had to settle for a certain subset of parameters, which was based on an 'educated guess'.

As mentioned in chapter 1.3 'Objectives and research questions', the down side with simulations is that some real world phenomena, which are not coded in the simulator, will not show in the results either. This depends on the coarse graining level (see chapter 3.4 'Complex adaptive systems') of the simulator. In other words, it means the level of details down to which the reality is described. Although the simulator uses the same algorithm as the real networks, some details have not been implemented. For example, fast fading is not simulated in the handover algorithm. Also the power control dependent part of the power budget handover algorithm has not been implemented. This probably inhibits some real life interdependencies, at least between the power budget handover and power control, from showing in the simulator. We can deduce from this that the parameter interdependencies have not been considered a matter of interest in cellular network design. This also indicates that the complexity issues have not been considered in the research and design of these systems.

In Figure 29, we can see the interdependence of a couple of power control parameters. Actually, the power control limits are defined by lower and upper thresholds (**`pcLowerThresholdsLev`** and **`pcUpperThresholdsLev`**), but in this simulation result we see the power control centre point and the window height. The lower threshold is the limit below which the measured power level must fall before the transmission power at the other end is increased. The higher threshold is the limit, which the measured power level must exceed before the transmission power at the other end is reduced. As we can see, the effect of window size depends on the absolute level down to which the window is adjusted.

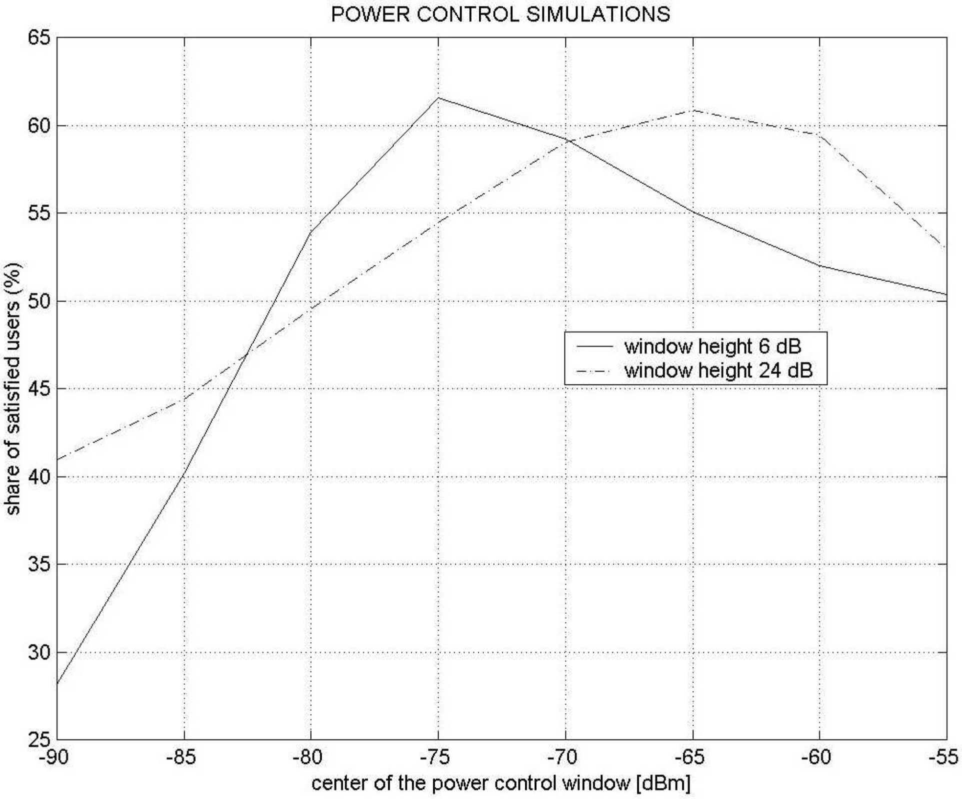


Figure 29. Interdependence of power control absolute level (window centre) and power control limits (window height). The absolute level of power control window has been stepped from –90 dBm to –55 dBm with 6 dB window height, and then again with 24 dB window height.

In Figure 30, we can see the interdependence of two parameters related to power budget handover. The effect of **`HO_margin_PBGT`** has been simulated from 0 dB to 16 dB with two different values of another parameter **`HO_period_PBGT`**. The effect is studied with the aid of percentage of 'satisfied users', defined as the number of connections that have average Frame Error Rate (FER) of <1%. The up most line represents the value of 4 SACCH-multi-frames (approximately 2 seconds) in parameter **`HO_period_PBGT`**. The next curve represents the value of 16 SACCH-multi-frames (approximately 8 seconds) in parameter **`HO_period_PBGT`**. The parameter **`HO_margin_PBGT`** defines how much better a new base station candidate must be so that a *power budget handover* to that cell would be possible. **`HO_period_PBGT`** defines how often the condition is checked. As we can see, the effect of the margin changes, if we change the period. There is also the effect of the same parameters on dropped call percentage (the down most two curves) in addition to the satisfied user percentage.

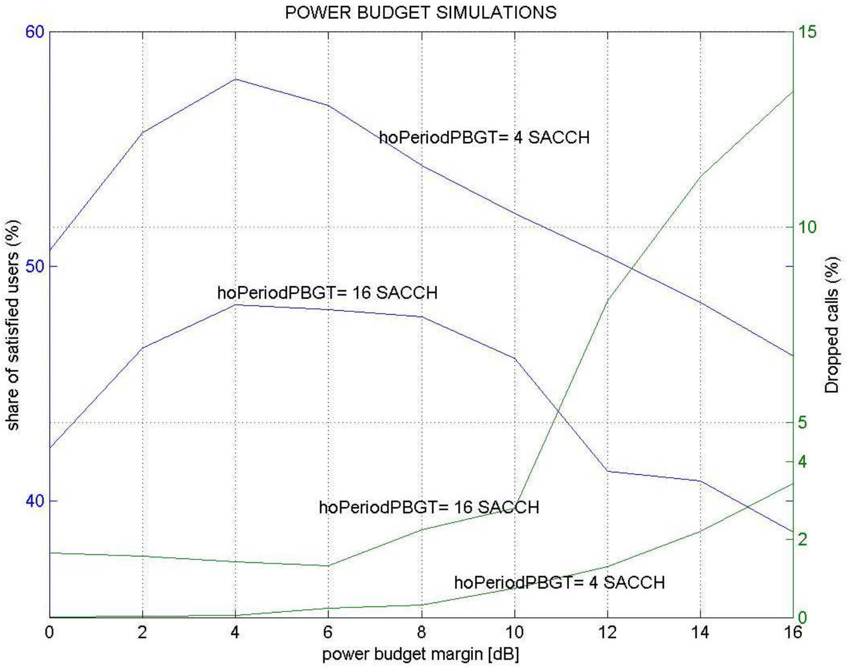


Figure 30. The effect of **`HO_margin_PBGT`** and **`HO_period_PBGT`** measured on number of satisfied users and dropped call percent.

Like this example indicates, one has to understand the phenomena very well in order to be able to adjust all the individual parameters so that the performance of the whole network is in optimum. Actually, it is more than likely that optimum performance can never be found considering dynamics of the network, number of different parameters (more than 200, most of them base station specific, some even TRX specific) and the interdependencies between some of the parameters. In a complex system, the number of fitness hills is so huge that it is impossible to find the global optimum in any manner. Even finding the excellent ones can be too overwhelming for human actors. It should not even be necessary, as the technology can be designed to find the excellent settings autonomously.

6.5 Typical working groups of a network operator

In this sub chapter, we shall define a network operator and describe some typical work groups of a cellular network operator to give an idea of what kind of functions are typically required in every day technical operations. We begin by giving a definition for network operator (TeleManagement Forum 2000[160]): "Network Operator is an organization that operates a communications network, network or data services capability, acting basically as a wholesaler. A network operator is a service provider. A service provider may provide the network operator role or may subcontract this role". "Other providers include service providers and network operators who are subcontracted by the customer's service provider to deliver the original customer request fully or as a component of the request." In other words, the network operator owns a network, which it uses for providing service delivery from the content providers to the end customers. The network owner either operates the network itself or it is also possible that the network owner subcontracts the operating work. The network operator can also be a service content provider itself.

As discussed before, there are very many different user groups operating complex technical systems, for example, cellular networks. Some of these groups utilise many additional tools besides NMS, designed either in-house or by 3rd parties, to cope with their work. Some groups do not even have anything to do with the user interfaces of the system itself. Frequently, the groups have to synchronise their work with other groups and be in real-time contact with each other to accomplish certain tasks. Only the NMS group of the various groups presented in Figure 31, really works with the main user interface of the system in their every day work. Radio network planners might do that occasionally while setting the BSS-parameters, if it is in the operator's working policy.

Many times, the NMS people use the command based MML[a] control to directly access the BSS[b] without the graphical user interface provided by the NMS. The NMS-users also quite frequently ask

---

[a] Man-Machine Language, text-based command language with a standardised structure, designed to facilitate direct user control of a system.

[b] Base Station Subsystem. This subsystem forms the radio network.

for help from the OMC[a]–support group. This group has provided many tools for even the most common work duties, since the network suppliers have failed to produce usable tools for the tasks. The help from the mentioned group is also frequently needed in accomplishing the work tasks. We shall briefly take a look at the most typical work groups of network operating in the following.

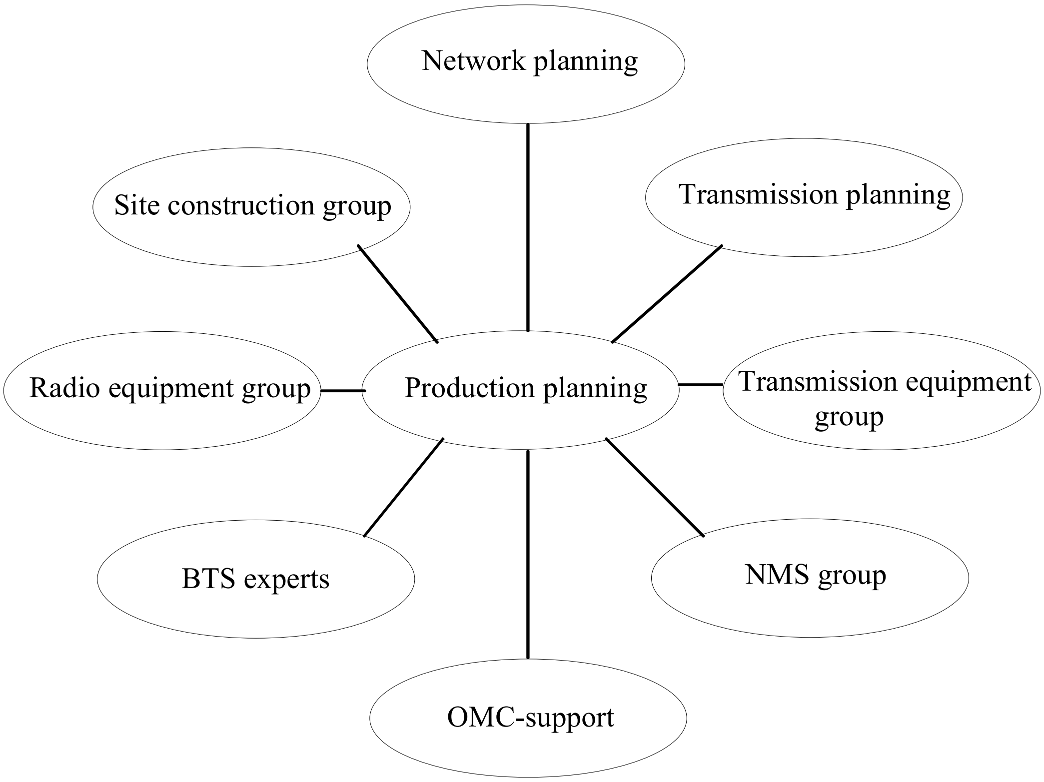


Figure 31. Typical groups in operator's organisation required for network operations. Network alteration schedules are created by the production planning group, but almost all groups are frequently in contact with each other.

Radio network planning engineers do the coverage, capacity, frequency and parameter planning of the network, as well as optimise the numerous BSS-parameters. They have a planning tool for this purpose, which can be their own (designed in-house), a third party tool or sometimes also delivered by the network supplier. There is an interface in the NMS, which can be used for transferring network plans in ASCII[b] form (text) between the planning system and NMS (the network). Some operators have also given the network planners an access to NMS, so that they can alter the BSS-parameters directly. Even in this case, the other planning data in addition to parameters, must be fed through the ASCII-interface. The network planners also need to access the statistics gathered by the

---

[a] Operation and Maintenance Centre, maintenance centre of the telecommunications network to which the operations and maintenance of network systems can be centralised.

[b] American Standard Code for Information Interchange, comprises auxiliary characters, numbers, letters of the Latin alphabet and special characters.

NMS, and hence have to transfer data in the opposite direction. After creating the network plans, the planners send the equipment lists to *production planning*, where a detailed building schedule is created.

Transmission planners plan the transmission network between the BTSs[a] and the BSCs and less frequently the links between the MSCs[b] and the BSCs. They create the connection orders for the *transmission equipment group*, who implement the connections. The transmission planners also design the connection alterations for different kinds of switchovers, where the site configurations are changed, for example BTS equipment changes. Usually they either have a tool delivered by the network supplier or they are using a third-party tool. Transmission planners need information from the radio network plan in order to plan the transmission links.

Transmission equipment group is working in the field visiting the transmission equipment sites and implementing the connections planned by the transmission planners. They need information from the transmission plan in order to make the required connections. They are also bound by the building schedule designed by the *production planning*. The connections must be ready when the BTS commissioning starts on a pre-planned date.

NMS group takes care of all the every day operations in the control room. Their duties are, for example, network surveillance (BSC/BTS alarms), network reparation/recovery, various alterations and construction/disassembly. A great deal of the work consists of assisting the other working groups by giving information, doing small tasks and entering data into the system. There is more about the duties of the NMS group in the next chapter.

OMC support group takes care of the NMS-servers and that the applications are working properly. They create the solutions when the system cannot provide some functionality or the system usability of the provided functionality is not acceptable. Their help is also frequently required in normal operations.

BTS experts design different BTS configurations and provide HW data for the base stations. They also help the *radio equipment group* and *site construction group* in problem cases.

Radio equipment group is, like the transmission equipment group, working in the field and visiting the BTS sites. Their duty is commissioning and maintaining the base stations. They are also bound by the building schedule, since the base stations must be taken into use on a pre-planned date.

---

[a] Base Transceiver Station.

[b] Mobile services Switching Centre.

While doing their work, they have to synchronise their efforts with the *NMS group*, as they cannot accomplish the tasks without help from the NMS-users. See also the BTS commissioning process example in chapter 6.8.4 'Commissioning'.

Site construction group builds up the sites, installs the BTSs and takes care of the software package installations and updates. Sometimes they have to communicate with the *BTS experts*.

Production planning group creates, based on radio network plans, the network building schedules, where all the site works and commissioning activities are stored. This database is called the Equipment Information System and it is not connected to the NMS. All the network extensions follow this schedule. This group also orders the BTS equipment and antennae for the new sites.

This list of groups is typical and by no means exhaustive. In addition to the described groups, there are still some others, for example, the *HW transport group*, which takes the materials and BTS equipment to the sites for site construction. Different operators have also divided the tasks in a little different way.

6.6 Network management group

The network management personnel working in the network control room (see Figure 32) are the main user group of the NMS. In some operators' organisation, also other groups, for example the network planners, might be using the NMS to some extent when modifying the radio parameters. The users in the control room can be divided into different groups by certain factors, such as their role in the organisational structure e.g. performance management personnel, network monitoring personnel, network configuration and maintenance personnel. Another possible division would be by privileges, e.g. standard users, super users and administrators. The former division naturally differs from operator to operator.

Many times, the NMS-users have two different computers. One Unix-workstation for running the NMS and another one, a PC, for using the local area network (LAN) and running the electronic GSM technology manual by the network supplier, Lotus Notes and some in-house developed applications. For example, when creating new BTSs, the users might fetch the BTS information from connection order files (Equipment Information System) by using the PC. That information is then utilised in NMS for creating the base stations.

Figure 32. A typical view to a network control room.

In the following list, there are some typical work tasks that the NMS users perform in the control room, although almost all of them require co-operation with other groups to be completed.

1. BTS commissioning
2. Switchover (Equipment change)
3. Changing BTS name, CI[a] or LAC[b]
4. Adding new neighbours to a BTS
5. Adding or removing TRXs[c]
6. OMU[d]-link and TRX route changes (PCM[e]-and time slot change)
7. Splitting a BTS under two BCFs[f]
8. Merging two BTSs under the same BCF
9. Transferring BTSs from one BSC to another
10. Transferring BSCs (with the radio network) from one MSC[g] to another (1-2 times a year)
11. Radio network monitoring

---

[a] Cell Identity
[b] Location Area Code
[c] Transmitter/Receiver
[d] Operation & Maintenance Unit
[e] Pulse Code Modulation (transmission link)
[f] Base Control Function. Handles common control functions, e.g. frequency hopping, within a BTS. On a sectored BTS site, there is only one BCF, which takes care of all common functions of the BTSs on this site.
[g] Mobile services Switching Centre

Every operator has to perform these tasks, and if the suppliers could offer good and efficient ways of accomplishing the tasks along with the required tools, the operators would most probably use them. At least the interviewed persons did not have anything against that.

6.7 Network management system

In this sub chapter, we shall give a brief introduction to a network management system to give an idea what kind of functionality it contains. Hi-Tech Inc.'s NMS contains a large set of applications to govern the network operations. They are used to monitor and control all Hi-Tech Inc.'s network elements. They allow for network troubleshooting, billing information acquisition, network reconfiguration and obtaining information on long-term trends. These functions can be summed under different categories, three of which will be more closely studied here with the aid of the information gathered from user's guides of Hi-Tech Inc.

These categories are *Fault Management*, *Performance Management* and *Configuration Management*. The purpose of the following sections is to briefly introduce, from the supplier's point of view, the different features of the NMS concerning network monitoring and maintenance duties. However, as the focus of the research is on the operator view, we shall not give any detailed descriptions of the supplier's NMS, nor the way the supplier understands the network operating.

6.7.1 Fault Management

This section gives an overall picture of Hi-Tech Inc.'s NMS-tools that can be used for *Fault Management* in the network (FM BOPP 1998[161]).

In a normal situation, alarms come in to the NMS in random order and from all parts of the network. Monitoring personnel have a full responsibility for processing the alarms. This job consists of sifting through the enormous amount of information presented by alarms deciding which of the thousands of alarms shown are important. It includes making associations between alarms and trying to decide which ones are possibly caused by the same fault and grouping those alarms together, as well as finding solutions for the problems. Fault monitoring consists of four different areas, each of which contains one or more tools (SW applications) or toolkits (see Table 6).

Table 6. Fault monitoring tool overview.

Online monitoring and investigating	7 tools
Alarm reduction	2 tools
Offline analysis of alarms	1 tool in PC
Alarm collection from 3rd party elements	3 toolkits for integration + 1 tool

Online monitoring and investigating contains, for example, the Top-level User Interface (TLUI), which contains graphical views of the network with network elements represented with symbols. One of the main functions of the TLUI is to show the alarm situation in all managed objects.

Alarm reduction contains, for example, the Alarm filtering and reclassification tool. The biggest problem facing network monitoring personnel is the fact that the flow of alarms can be huge. Alarm filtering allows the user to block certain alarms from the NMS. They can be blocked from the views of the alarm handling applications, or entirely so that they are not entered into the database. Alarm reclassification lets the user assign a different class to an alarm, so that it is e.g. more visible than it was originally.

Offline analysis of alarms contains a toolkit called Alarm statistics in PC. It provides tools to retrieve alarm data from the NMS database and generate graphical or textual reports on the basis of that information. The feature is built on Microsoft Windows and uses Microsoft Excel for data processing.

Alarm collection from 3rd party elements offers possibilities for integrating other manufacturers' equipment with the NMS. The idea is that alarms from 3rd party equipment can be collected, viewed and stored in the NMS along with the other network alarms.

6.7.2 Configuration Management

This section gives an overall picture of the NMS tools, which can be used for *Configuration Management* in the network (CM BOPP 1998[162]).

The BTS *Hardware* Configuration Management is designed to provide tools for monitoring the hardware configurations of the base stations remotely from the NMS site. It is implemented by two graphical user interface applications, one for viewing BTS hardware (HW) data and another one for locating hardware units in the hardware database, as well as generating reports on the basis of search results.

With the BTS *Software* Configuration Management, one can remotely download and activate a new or upgraded software package in one or several BTS sites. It consists of three graphical user interface applications. The first one is a database containing data about existing software packages in the network elements. The purpose of this application is to maintain an up-to-date view of the software configuration of the BCFs. With this application, one can also compare for common or different modules in the software packages. The second one is an application, which can be used for background downloading a software package to a BCF and then activating the package, either immediately or at a later date and time. In addition to this, it allows one to examine the progress of these operations. The third one is an application, which allows one to manipulate the software packages in individual BCFs. With this application, one can e.g. activate the package, swap the status of two packages and delete software packages from a network element.

6.7.3 Performance Management

This section gives an overall picture of the NMS tools, which can be used for *Performance Management* in the network (PM BOPP 1998[163]).

The purpose of any performance management activity is to collect data, which can be used to verify the physical and logical configuration of the PLMN[a] and to localise potential problems as early as possible (ETSI specification ETS 300 615[164]). In addition to monitoring network faults, the operator needs instant information on how the network performs from a mobile user's point of view. In this respect, the *quality of service*, as observed by the subscribers, is one of the most important areas to be monitored. Network performance can be evaluated based on *key performance indicators* (KPI), such as the traffic channel drop call ratio. The operator can use the data collected from the network by the NMS and calculate various measurement results (PM BOPP 1998[163]). Measurements are stored in separate tables in the NMS database – usually one table consists of one measurement. In addition to these tables, there are also long-term measurement tables on traffic, resource access and handover measurements. The long-term tables contain daily, weekly and monthly summaries that are calculated daily from the actual measurement results. When the network is smaller, it is possible to have more measurements running simultaneously. When the network grows, the data amount gets bigger and bigger.

The operator's daily routines include for example, monitoring of the faults and network performance, locating and analysing faults as well as corrective actions. The Performance Management applications help the operator to detect, for example, service-related problems by providing means to analyse measurements dealing with call attempts, dropped or blocked calls, and handover failure rates. Performance monitoring can be divided in several categories, each consisting of several tools (see Table 7).

Table 7. Performance monitoring tool overview.

Reporting and post-processing	4 tools
Monitoring	2 tools
Troubleshooting and online monitoring	3 tools
Administrating	2 tools

The *reporting and post-processing* applications allow the user e.g. to retrieve, process and visualise the measurement data collected into the NMS database. It is also possible to transfer the data to a PC and do these things with Microsoft Windows tools.

Monitoring contains two applications. One is for automatically verifying the new measurement values and defined thresholds and generating an alarm if the threshold criteria are fulfilled. Another one is for retrieving data from the NMS database, as well as generating monitoring reports and graphical presentations on the basis of user-specified Quality of Service (QOS) indicators.

Troubleshooting and online monitoring applications are for further investigating the problems found e.g. by using the monitoring applications. The applications can be used, for example, to collect data from the traffic channels, which use a certain interference band, or to obtain useful information of the network performance after certain adjustments.

Administration contains functionality, which allows the user to create, start, modify and delete BSC measurements from a centralised location using a graphical user interface. In addition, it enables the user to view the number of measurements stored in the NMS database.

In the following chapter, we shall study how the network operators perceive in practice the functionality and system usability of Hi-Tech Inc.'s technology when operating their networks. As the NSS part of the network was left out of the scope of this study, we are concentrating on the

---

[a] Public Land Mobile Network. A mobile network for the specific purpose of providing land mobile communication services to the public.

operation of BSS. Since OSS has a very important role in operating the BSS, we shall also see the operator's view to utilising the NMS (Network Management System) and some other tools for management in their operation.

The author has applied the theoretical framework for interpreting the empirical data. To point out some of this interpretation to the reader, the text contains comments by the author. Those comments are typed with a different letter type *(Kaufinn)*, to distinguish them from the rest of the material.

6.8 Operator A

Operator A is an experienced operator with well established operating processes and they are not very actively searching for improvements in the processes. They have a medium size network.

The cases in this research are operating tasks of operator A. In the case selection, the operator was first asked to estimate when they have a busy period, during which it would be likely to see the most common tasks in practice. The observations were then scheduled to cover that period. All the relevant happenings within the period were observed and recorded. As we wanted the cases to represent the most common tasks, the cases were selected afterwards from the observed events. To accomplish the tasks, the operator was also using many applications they had created in-house. Since we do not want to imply the operator in question, we use coded names, such as Own_application_1, instead of the real application names.

List of cases

Case 1: Creating a new cell
Case 2: Setting the neighbours
Case 3: Network planning
Case 4: Commissioning
Case 5: Splitting a BTS into two cells
Case 6: BSS-splitting
Case 7: Switchover
Case 8: Fault management

We shall now study the cases in practice one by one to see how the operator personnel copes with the operational tasks by using various applications of the NMS, the low level command language (MML), 3rd party tools and the tools of their own. After each case, there is a short summary of the observed actions and some improvement suggestions. The purpose of the suggestions is to show how easily the present, rather high effort could be reduced. This also proves that the present system usability cannot be very high. After the last case description, there is also a common conclusion of all cases.

6.8.1 Case 1: Creating a new cell

Before a base station can be taken into use, it must be physically built and logically created in the network databases. The schedules for building and commissioning the new base stations are defined in the connection order list (Equipment Information System), which is prepared by the *production planning group* (see chapter 6.5 'Typical working groups of a network operator'). In addition to the NMS-users in the control room, there are many other people involved. For example, the *radio network planners* have created the radio network plan, where e.g. the neighbour cells are visible, and based on the radio network plan the *transmission planners* have designed the physical transmission links from the MSC (via BSC) to the base stations. Hence, the preparation works depend on the efforts of many interdependent groups.

The cell creation procedure is as follows:

1. Create the parameters for the BTS by using Own_application_1 (operator's own tool)
2. Bring the newly created file into NMS (Hi-Tech Inc.'s NMS)
3. Create the transmission links by using MML and connect them to the BTS (NMS tool)
4. Create a new BCF and send into the network (NMS tool)
5. Update the BSC-view and send the BTS data into the network (NMS tool)
6. Update the MSC database by creating the cell (MML)
7. Give the network planners the right to edit the view (Own_application_3)

An NMS user demonstrates the actions he uses for creating a new cell (a base station). The user states that he uses their own tools (created in-house) for some actions in the cell creation, for then he only has to give once the required parameter data.

He starts by creating the BTS-file with the operator's own command line based application by giving the name of a default parameter set as an argument. Default parameters are used at this point and the values are changed by *radio network planners* in a later phase if required. There are many different default lists and the user has to use the correct one in each situation. Particularly, in a multi-layer, multi-band network the number of default parameter lists becomes large. According to the users, it would be good if the cell creation/commissioning tasks could be simplified, since they are required quite often.

He creates the BTS-file based on the data, which was stored in the equipment information system by the *radio network planners*. ***This indicates interdependence of work groups, as the data must be stored by the network planners before the creation is possible.*** Next, he imports the file to the NMS by using a certain NMS-application (the network planning tool interface), which can be used to import ASCII-based data. Then he creates the transmission links (OMU and TRX links) by MML-commands with the aid of the data, which was stored in the equipment information system by the *transmission planners*. ***This indicates interdependence of work groups, as the data must be stored by the transmission planners before the creation is possible.***

The transmission planners have planned the routes already about a month before, because they have to reserve time for building the physical signal path. When the links are originally planned, this data is not, however, used to eliminate the manual link creation effort. The NMS-users must still manually create the links, even though the transmission planners have a tool from the same supplier as the NMS. Now the NMS-user connects the links to the BCF and TRXs by using an NMS-tool. There is another GUI[a]-based tool for creating and handling the transmission links in the Hi-Tech Inc.'s NMS, but due to its slowness the users prefer not to use it – except for mass destroying links. Now he creates the BCF-view and sends it into the network by using an NMS-application.

Next, he creates the BSC-views from the imported data by using another application of their own and sends the data into the network by an NMS tool. After creating the views, all the newly created BTSs are available for the other NMS users. ***The view creation seems to be a bottleneck, since the views are required before the network planners can modify the radio parameters. This kind of interdependence should be eliminated.*** The MSC-database is updated later when doing the commissioning.

The interdependencies between the tasks of different groups make the tasks more complex and slow down the work. According to the users, it should be possible to create the BTS by one person at one go, not sequentially with different groups as it is now. Also here, there are different ways of doing things. In some areas in the country, the network planners set the parameters (after the views are created in the control room), in some other areas the network planning data is read from the equipment information system and manually entered in NMS by the NMS-users. Both the network

---

[a] Graphical User Interface.

planners and the NMS-users expressed their discontent with the view creation/parameter setting dependence. Mutual dependence increases complexity by definition, as the K-factor increases.

"It would be good if the parameter data could first be stored there somewhere, so that it wouldn't need the view there first – or then the view creation should be made so easy that the network planners would care do it. Then, running the BTS data into the network and creating the link would be our only tasks."

Now the NMS-user creates the cell into the MSC database by issuing MML-commands. Finally, he gives the radio network planners the right to change the views and manipulate the data by using an NMS tool. According to the users, the benefit in using their own tools for the processes is less manual effort. If the Hi-Tech Inc.'s applications are used, then one has to drag the network elements one by one into the Network Editor and manually supply the required parameters for them. Whereas, by using their own applications, the user only gives the name of the default parameter set, and the parameters are automatically copied to the new BTS. Before they had their own application for the cell creation, he used two different Network Editor sessions for the task. There is also another Hi-Tech Inc.'s application for creating cells, but the downside with that is that it requires a reference cell, which can be difficult to find manually in a large network with very many different configurations. This is user-observable complexity (random complexity). It does not seem to be quicker than using two Network Editor sessions.

Assessment

There are many manual actions, which could be eliminated by process design and tool modification. The operator has designed additional tools for this task, for example the tools for creating the parameter file and updating the cell-views. MML is also used quite a lot. Obviously, the operator has not considered Hi-Tech Inc.'s tools very suitable for the tasks. The tools possess rather high random complexity and do not support independent, highly automated ways of working, which also reflects to the operating process. Interdependence between the work of e.g. radio network planners and NMS-group complicates the operating tasks. For example, the network planners cannot set the parameters before the NMS-users have created the BTS-views. Even if we accept the manual handling of the process, the data transfer between the planning system and the NMS is not fluent, because it requires either several format conversions or manual data entry. In addition, there are many different default parameter lists, which the network planners and NMS-users have to deal with. This adds the user effort and user-observable complexity of the task.

The users told that they do not have exact rules defining which tools should be used for any single action. This has lead to the situation, where different users use different tools in the same situation – some of them use Hi-Tech Inc.'s tools, some of them use the tools of their own. New users learn the operating actions from the older users, which means that the way of doing things depends on the source the action was learned from. Since there are so many tools, the use of which is not collectively defined per task, many users deploy quite inefficient ways of performing the tasks. The inefficiency increases the user effort and hence decreases system usability.

Suggestions for improvements

As the interdependency (K-factor) of the tasks of NMS users and network planners is rather high, it would be advantageous to reduce it. Particularly, the view creation process is a bottle-neck and should be eliminated. Also the user-observable complexity in the tools should be reduced. Since the work is done at a quite low level, metasystem transitions in the tools would be a good choice for reducing the user-observable complexity.

According to one user, a good system would create a new BTS, for example as follows. All the required parameters are directly stored in the NMS-database, when the network planner creates the plan. The tool would automatically create the new views, when the necessary data is available. The transmission links would be created directly to the NMS-database by the transmission planner when he designs the connections. Finally, the user would send the BTS into network, which would also trigger automatic updating of the BSC and MSC databases. However, this procedure would require well functioning interface between the planning tool and the NMS. Preferably, they should be using the same database. In addition, if the status of the new elements were automatically updated in the view, based on the actions performed at the base station site, then the NMS-users would see if a certain BTS is ready for commissioning. Actually, if the tasks and tools of the radio equipment group were slightly modified, they could also take care of the commissioning autonomously without any help from the NMS-users. The radio group's task could also be highly automated with the aid of their PC. The NMS-users would only be consulted in case something goes wrong. See the description of commissioning in chapter 6.8.4. All these actions would reduce user-observable complexity and operating effort thus improving system usability. Similar improvements could also be done in many other tasks.

6.8.2 Case 2: Setting the neighbours

The neighbour setting procedure:

1. Read the neighbours from the equipment information system.
2. Find the neighbours one by one in the cell lists (NMS-tool).
3. Open and copy the cells one by one for both directions separately (NMS-tool).

Or

1. Read the neighbours from the equipment information system.
2. Copy the neighbours to Own_application_4 and execute.

According to the user, setting the neighbours is not a simple task. The user has to find every neighbour from a long list (hundreds of cells), sometimes under another BSC. **This is user-observable complexity (random complexity).** There is also too much manual repetition of certain commands (File Open, File New). Consequently, the operator has produced their own tool for the task (Own_application_4). According to the user, it would be much easier, if the network planning data could be transferred in electronic format to the Own_application_4. Presently the data is manually copied from the equipment information system.

First, the user starts the task with Hi-Tech Inc.'s NMS tool. He is not familiar with the tool and soon he asks for an advice from his colleague. It appears that the user accidentally destroyed the already created neighbours while doing the neighbour definition. Desperately they try to restore the situation, but without success. The other user states that the tool was definitely to blame. An UNDO-function would be useful in a situation like this. The users try to destroy the deficient neighbour definitions that appeared in the NMS-database but not in the BSC by moving the definitions from one NMS tool to another. They begin to be quite discouraged, but they are not using the NMS documentation. They both agree that creating the neighbours requires too much effort, if it must be done this way – one neighbour at the time, by defining both directions separately. MML would be faster. The other user agrees by saying:

"This really hits rock-bottom."

The user starts to use the character-based Own_application_4. The users think that setting the neighbour data this way is faster and more reliable.

"Isn't it a little simpler, if it goes like this?!"

The user also thinks that he should be able to do more actions with the same tool. The less separate windows there are open simultaneously, the better. **User-observable complexity. This falls into the category of random complexity.**

"The first think that comes to mind from the user interface is that I should be able to do everything in the same window. That, at least in the beginning, confuses, what should it be done with? So that creating the view and the neighbours…it would be good if they could be done with the same… Anyway, this is the tool, which is used in normal operating…Top-level-UI."

An additional problem is the neighbour cell definition across OMC borders (see Figure 33). If the adjacent BTSs are in different OMCs, the neighbour cell definitions must be done in both OMCs separately. Although the user can open another session for the other OMC with the same NMS, he still has to enter the data twice, once for each OMC. The OMCs do not communicate and update the situation if the neighbours are defined in one OMC only.

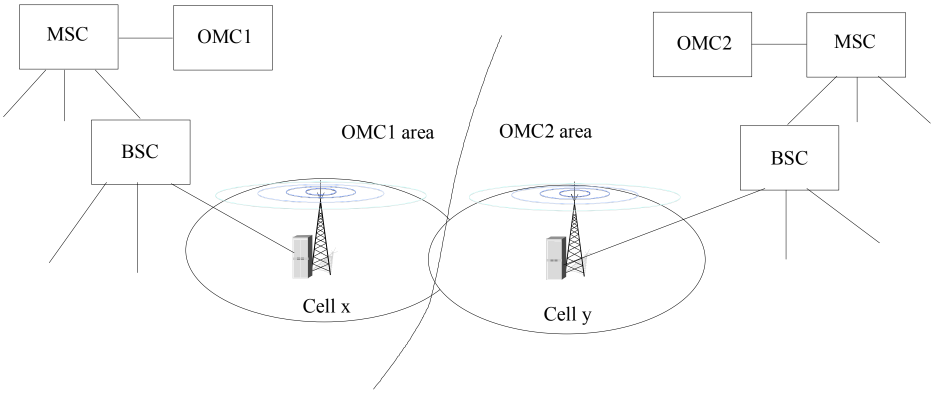


Figure 33. When cells x and y are defined as neighbours, the definitions as well as all parameter modifications must be done in both OMCs. This increases operating effort and probability of errors.

Assessment

The users think that using the NMS-tool requires too much effort, as they have to pick all the neighbours one by one from long lists and repeat the File open/File new –sequence for each of the neighbours. The neighbour definitions must also be done separately for both directions. In addition, the OMC-border areas cause extra work, since the definitions must be done twice – separately for each OMC. The users feel that they have to use too many applications for the task, and hence they have very many windows open on the screen simultaneously. This random complexity increases user-observable complexity of the task. The amount of manual effort for performing the task is also quite extensive and mistakes are difficult to mend.

Suggestions for improvements

Reduction of user-observable complexity would also be required here. The neighbour setting procedure is actually redundant, since the neighbours are already defined in the network plan. The system should utilise this existing information and spare the users from the manual effort of doing the same work again.

Since the neighbour definition must be done once in the planning system, a good and fast way of doing it would be as follows. All the cells would be displayed geographically on map. The user would switch the 'Neighbour Definition'-mode on, and the cursor would define a circle with the wanted BTS as the midpoint. By moving the mouse, the user could change the radius of the circle. When the wanted neighbour cells are inside the radius, he would click the mouse button and the selected BTSs would immediately become defined as neighbour cells. As a default, the definitions would be two-way definitions, but the user could change the default. If the user is not quite satisfied with these neighbours, he could then add or remove some cells one by one. The communication between the OMCs should enable automatic data update across the OMC-borders, when the data is entered once.

6.8.3 Case 3: Network planning

The planning procedure starts when the subscriber data and forecasts are delivered to the network planners. Then the planners create coverage, capacity, frequency and parameter plans for the required areas. After creating the network plan, the planners deliver some data, such as antenna lists, to the construction unit in *production planning*. There, a short term building plan is created and stored in the equipment information system. In addition, the orders to the antenna suppliers, as well as internal work orders are made here.

Network planners are also involved in radio network parameter optimisation. Initially, the NMS-users load a set of default parameters, which the network planners optimise later based on the measurement data. The planners gather the measurement data in the field with the tool developed in-house and also use the statistics gathered by the NMS counters. An outside company (3rd party) has designed an interface between the network planning system and the Own_application_5 (the field measurement tool), so that the measurement results can be transferred to the planning system. These are not the only tools the network planners have to use, there are approximately 10 different tools in use plus the NMS-applications. ***This is user-observable complexity.*** After analysing the measurement statistics, the planners create a parameter optimisation plan, according to which the parameters are modified. After the modification, they have to verify the situation by checking the statistics from the NMS again. According to the network planners, it would be great, if the BSC-algorithms had less parameters (*This is reduction of user-observable complexity*), or the large amount of effort with the parameters could somehow be avoided.

The network management system (NMS) and the network planning system have always been separate systems, even when delivered by the same supplier. This means that there is an interface, through which the planning data is loaded into the NMS and the network statistics are loaded into the planning system. This always slows things down, as everything must be manually loaded through the interface and not have all the data readily available in both systems.

The parameter optimisation is a demanding planning phase. The number of different parameters is large and many different default lists must be maintained because of the modern multi-layer/multi-band networks. One dilemma in the optimisation is that there is no logbook (history file) containing the parameter value changes, reasons why those alterations were made and when. *This also tells about the dynamics of the operating landscape.* Unless the reasons for the changes are known, it is risky to change the values. As it is not possible to keep that log in the NMS, the planners have to use other means for it. In addition, the same planners do not stay on the job forever, but change about every three years and transferring the tasks to the new planners is difficult. The new planners, many times, do not know why certain parameter values are as they are. It can also happen that the new planners dare not touch them, because they feel uncertain.

A new cell-view is made every time a new BTS is created, and presently the views are created by the NMS personnel. The views are required before any radio network parameters can be set to the network. *This is interdependence of work groups.* According to the planners, either creating the views and setting the parameters should be independent of each other or the view creation should be automatic. The operator has circumvented the serious shortcoming of Hi-Tech Inc.'s equipment (serial view creation and parameter setting) by developing a commissioning table, where the network planner can store the parameters before view creation to wait for commissioning.

Creating the neighbours, particularly across the OMC-borders, is painful. Information is not conveyed to other OMCs, and hence all the parameters at the border must be defined from two different OMCs (double amount of effort). All this is handled manually.

Assessment

Doing, for example, parameter optimisation involves a lot of manual effort and many different tools. Furthermore, this kind of work requires high level of skills and a lot of experience, since the planners have to understand all the complex features of the radio network in practice. Actually, they also have to create the understanding of how the parameter changes affect the behaviour of the system, since even the designers of these features do not completely understand the underlying phenomena, but instead have created a rich set of parameters so that the users could somehow tune the system when running it.

Since there are so many different tools, the interfaces between them cause additional problems, which the operator has to solve somehow. Even the tools of the same supplier do not always play well together. For example, the interface between the network planning system and the NMS is not very convenient.

Particularly, Hi-Tech Inc. has created a complicated BSS with very many technical features and manually optimised low level parameters. The operators try to avoid the parameter optimisation by using default parameter lists, but this does not solve the problem completely, some optimisation is always required. The users would like to reduce the number of the parameters, even by reducing the functionality of the system. According to them, there are some useless features in the system anyway, since the suppliers never remove anything. As the NMS does not contain any history file for the parameter changes, the planners have to maintain this data elsewhere. Knowing the reasons for the past parameter alterations is important when deciding about new values.

As pointed out already in Case 1, the dependence of view creation and parameter setting is a bottleneck. The operators have to do extra work to circumvent the restrictions set by the tools. The same approach is visible in many places: It requires a lot of preliminary planning; many things must be done either serially or synchronously, since the independent tasks are not enabled. The network planners also considered the neighbour cell definition across the OMC-borders causing too much effort (see Case 2 in chapter 6.8.2).

Suggestions for improvements

One rather easy solution to the parameter change logging problem could be a web-browser application connected to the OMC-database. This application would be containing a cell-based logbook and a link to the parameter descriptions explaining the function and purpose of the parameters.

The interface problem with the planning tool and the NMS could be solved, if the tools were using common databases. Then it would be easy to achieve the automatic view creation based on network planning data, which was mentioned earlier.

As the network planner's work contains many complicated tasks with many separate tools, reduction of user-observable complexity would improve the situation. Particularly this domain could use a metasystem transition to avoid the low level work with the numerous manually tuned parameters. To enable controlling the network on higher level, the low level complexity should be hidden by architecture. According to the principles of complex systems in nature, the complexity in lower layers should be handled autonomously. This would be a good application field for another natural principle, namely complex adaptive systems. When using this approach, the operators would only give some high level guidance to the system, letting the autonomous lower layers take care of adaptation to the environment.

One way of reducing the high complexity and manual effort, is to redesign the radio network algorithms so that the functionality is maintained but with much less parameters. Since nothing has ever been discarded or redesigned, some of the functionality is obsolete and could be removed in the redesign steps. Another, more radical possibility when redesigning the radio network architecture, is to create adaptive systems without manually tuneable parameters (refer to chapter 3.4 'Complex adaptive systems'). The latter possibility would take a few years to design and implement, which means that some smaller evolutionary steps are required in the mean time. In any case, a fully adaptive system would be worth researching, since it could enable network performance closer to optimum and save a lot of optimisation effort from operators.

6.8.4 Case 4: Commissioning

The commissioning procedure:

1. Load the HW-data (radio man)
2. Connect the HW-data to the BCF (NMS-user)
3. Activate the BTS (NMS-user)
4. Check the alarms (NMS-user)
5. Run the tests and measure the transmit power (radio man)
6. Check that the BTS conveys traffic (NMS-user)

When the commissioning starts, the NMS-users have already created the BTS with the aid of their in-house-developed tools, NMS-applications and the concurrently updated connection order list in the equipment information system (developed in-house). This was described in Case 1. After this, the NMS-user defines the neighbours (see Case 2 in chapter 6.8.2) and sets the parameters based on the data defined by the *radio network planning engineers*. Due to the problem with the dependency of the view creation and parameter setting (see Case 1 in chapter 6.8.1), the operator has started locally in this area the practice that the NMS-user creates the views and sets the parameters. This process has the down side that the NMS group can easily be overloaded with work. There are simply too many time consuming manual operations. In some areas, creating the neighbours and modifying the parameters are done by the radio network planners directly after the NMS-users have created the views.

Next, the NMS-users do the commissioning in co-operation with the *radio equipment group* (field personnel). **There is a clear interdependence between the tasks of these groups.** The commissionings are marked in the connection order list. The whole commissioning procedure is based on this list, which is edited by one user at the time via the local area network.

The NMS-user opens the BTS created earlier. He switches between TLUI and MML wondering which one to use. With MML, he can see everything precisely, on the other hand everything is visible simultaneously in the graphical user interface (TLUI). The field man telephones the network control room to start the procedure. If the BTS has not been 'on the air' before, also the HW-data must be loaded and activated. The HW-data describes the configuration for a certain kind of BTS hardware and it is prepared by the *BTS-experts*. Usually, standard configurations are used, but sometimes it is necessary to create new ones. According to the users, the number of different, already existing HW-configurations is already too large and there is need to reduce this quantity. Usually, the HW-data is run by the field men locally at the base station site or occasionally by the NMS-users from the control room. The user is wondering which HW-data to use in this case. He asks for an advice from his colleague, and then tries to take up the BTS to see whether the field men have already run the HW-data. After a while, the field man calls and verifies that the correct HW-data has been loaded. Now the NMS-user connects the HW-data file to the BCF and activates the BTS. **The metasystem transition principle could eliminate many low level manual operations and interdependency.** There seems to be an incomprehensible inconsistency in the BTS-software: The HW-data must be set passive with an MML-command when one wants to activate it.

The field man also wants to know if there are alarms in the BTS. The field personnel have a terminal they can connect to the BTS and check the alarms, but usually they ask the NMS-users about the alarm data. The NMS-user starts monitoring the alarm situation of the network element by using MML. The user is in telephone contact with the field man and enters the same command several times when polling the BTS status. There are alarms coming from the BTS (No connection to the TRX…). Obviously, there is some kind of a problem and the NMS-user gives a reset to the BTS – this is usually the first thing to do when problems arise. If it does not help, then the next thing to do is to check the settings. This consists of link data checking in the control room and a branching table check at the BTS. As a last resort, if these procedures do not solve the problem, the TRX is changed and a new attempt is made. In this case, the reset clears the problem and the task can be finalised.

Next, the field man tests all the time slots with test calls, tries an emergency call and measures the output power of the TRX. When the BTS is running and there are no alarms, the NMS-user checks with MML-commands that there is traffic through the new BTS. Finally, he informs the field man that the commissioning was successful. When the field man has finalised his task, he marks the status in the connection order list. In practice, the radio group fails to update the status after the commissioning. Only about 20% of them are actually marked. According to the users, the status updating procedure should be automatic after the last action.

After the commissioning is finalised, the network planners monitor the system with the Own_application_6. This is a monitoring application, which utilises the counters of Hi-Tech Inc.'s system. Metrica is used for creating and visualising new information. Also TEMS (monitoring the signalling) and Own_application_5 are used in the field. The NMS-user tells that there are some tasks the system lets you do wrong. For example, the TRX numbering – everything must be done again if numbers for the TRXs are given in the wrong order. He remembers also when he accidentally removed a whole BTS when he wanted to remove a TRX. Then all the parameters, neighbour definitions and other data were also lost. He was shocked at the amount of destruction he caused by one press of a button.

Assessment

The operator has created flow charts for the operating processes. We can study the commissioning process with the aid of such a chart (see Figure 34) as an example. The figure is a flow diagram, which does not display all the details separately.

We can see that there are many actions, which depend on each other, and the procedure requires that some intermediary actions be taken by one group in order for another group to continue their work, for example the actions with HW-data. The different working groups have to make telephone calls continually to each other to get the others do some procedures or to get some information. This kind of working adds the complexity of the tasks and reduces efficiency. In addition, many tasks at BTS site are performed manually, even though it would be quite straightforward to automate the procedure with the field personnel's computer. The performed actions are also manually updated into the connection order list.

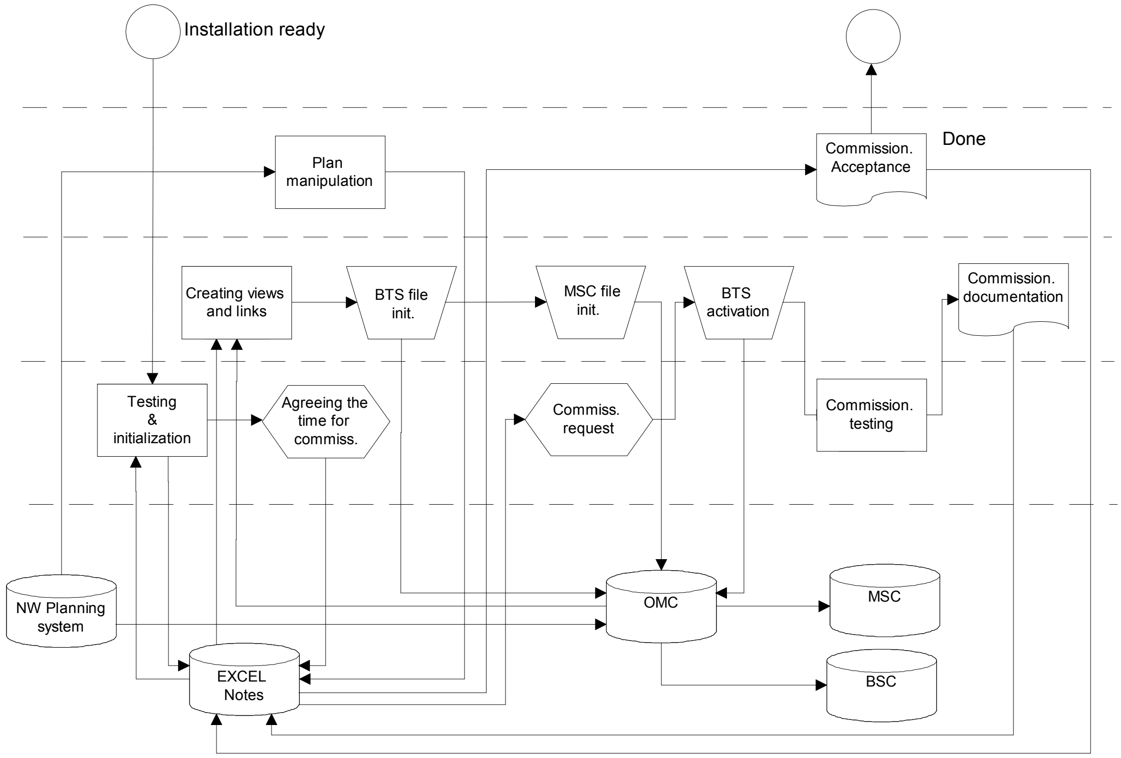


Figure 34. An example process (BTS commissioning). The dashed lines separate the tasks of different work groups. Source: Operator A's documentation.

Suggestions for improvements

Due to the high interdependency of NMS users' and field men's tasks, it would be very useful to reduce the K-factor of these tasks. In addition to this, the manual effort could also be significantly reduced by implementing a metasystem transition in the tasks. The first three operating tasks described separately in this document (cell creation, neighbour setting and commissioning) could be integrated into one task when implementing this metasystem transition.

The integrated commissioning procedure could perhaps be optimised the following way (see Figure 35). In this procedure, e.g. the field man can finalise the whole BTS commissioning task independently without asking any help from the Network Management Centre. He would load the HW-data, connect the HW-data to the BCF, unlock the BTS, check the alarms, verify the working by test calls. The status would be automatically updated after the last step. The whole procedure could be an automatic program in the field man's PC. He just connects the PC to the BTS and activates the correct program, which will complete the procedure. If there are alarms, then the field man would have to deal with them. Actually, it could be even so that the procedure would be run by the NMS-personnel in the control room and only if the procedure is not successful, the field group would pay a visit to the site.

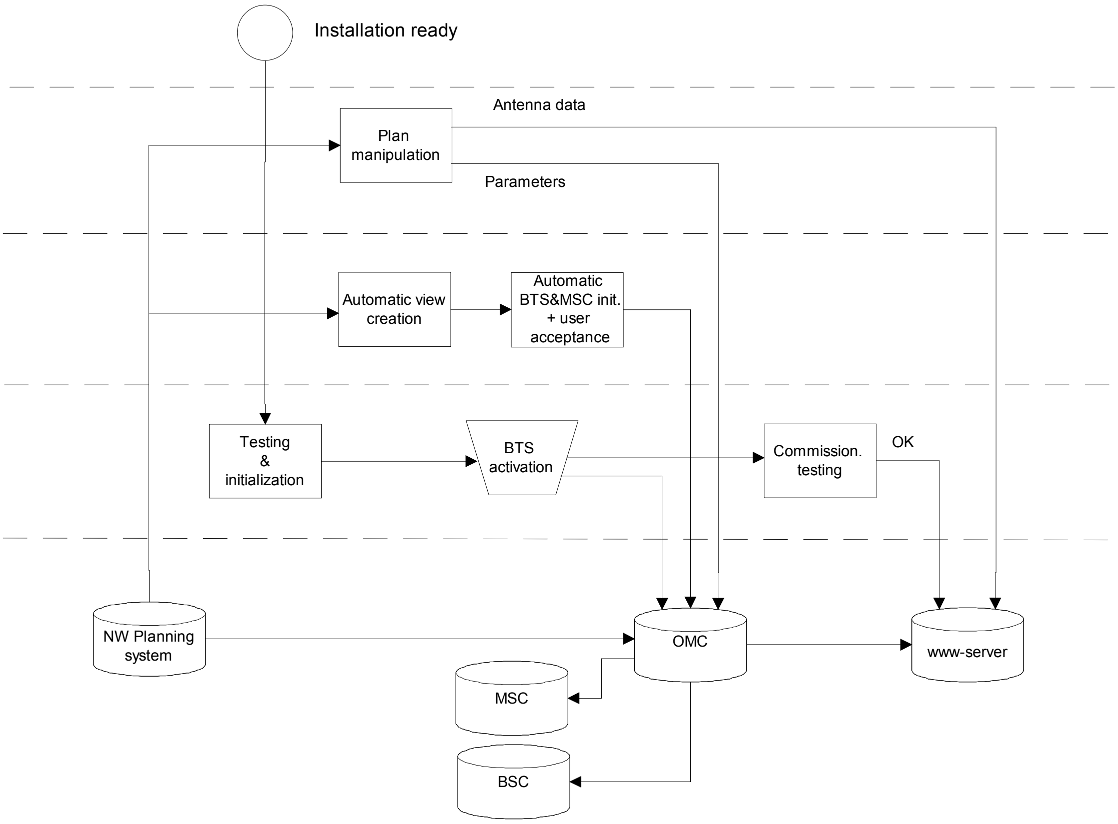


Figure 35. Improved process (BTS commissioning). The dashed lines separate the tasks of different work groups. Author's suggestion.

The views would be automatically created from the network planning data, which are stored in the planning tool. The transmission links would be directly created into the NMS-database by the transmission planner, when he designs the connections. The MSC[a]-database would be automatically updated based on the existing information already in the system. The new procedure would change the tasks of the field personnel, but since the procedure could be highly automated, it is no problem to the workers. For cases of malfunctions, they would require some additional training. It also requires some alterations in the network elements/tools. For example, the field personnel would need some kind of a remote tool, which they could use to access certain limited applications in the NMS, e.g. locking and unlocking a BTS.

---

[a] Mobile services Switching Centre.

6.8.5 Case 5: Splitting a BTS into two cells

The splitting procedure:

1. Contact OMC-support[a] (NMS-user)
2. Lock the network elements (NMS-user)
3. Create new BCF in the view (NMS-user)
4. Create new OMU-link and send to the network (NMS-user)
5. OMC-support transfers the BTS data to the new BTS
6. Add the new BTS data (CI and name) to the view by using NMS and send to the network (NMS-user)
7. Update the BTS-data and unlock the BTS in the MSC (NMS-user)
8. Inform the OMC-support about the possible TRX numbering changes (NMS-user)
9. Take backup copies of the TRX data, destroy the links and create new links (NMS-user)
10. Re-run the HW-data into both BTSs, since the configuration changed (field man or NMS-user)
11. Connect the HW-data to the BCF (NMS-user)

Cell splitting is a procedure, where one cell is divided into two cells, increasing in this way the total number of cells. Occasionally this is required when increasing the capacity of the network. In this case, the users are splitting a so-called duo-BTS into two separate BTSs. A duo-BTS has two cells, but it has been installed in one rack only and it has one BCF. When more capacity is required, the operator divides the cells into separate BTSs with dedicated BCFs. A new rack has been built for the new BTS and also a couple of new TRXs have been installed in there. Since this is an educational case for another, younger user, he is writing notes for himself as the other one guides him through the procedure.

The user checks the alarm history of the BTS by MML before locking it. He states that it is not a good idea to use the graphical tools of the NMS for this kind of duties, because they are slower than MML. The reason is that there is no way to lock the whole base station at once, but first the user must find the BCF, the BTS and all the individual TRXs and lock them separately. ***This is user-observable complexity (random complexity).*** Also, the graphical application does not display anything when started, but the user must open a certain file, which defines the data to be displayed.

The user calls the OMC-support to inform them about the split, and asks them to backup copy the BTS data. After this, the OMC-person stores the data and the NMS-user destroys the BTS. The

NMS-user creates a new BCF with an NMS-tool. This is done, because there is only one BCF in the duo-BTS and the new, separate BTSs both require a BCF. Then he creates the OMU-links for the BTSs with MML and sends them into the network. Now the user calls the OMC-support and asks them to transfer the BTS data from the backup file to the new BTS. Next, the NMS-user gives the new BTS a name and a CI by using Network Editor and sends the data into the network. After this, the NMS-user updates the MSC-database by creating the new BTS in the database and unlocking it. **As one can see, the task requires very much user effort, since no automated tools exist.** Now the users want to verify that the BTS is open.

"Check for example from the [TLUI] Mode that it opened."

For some reason, the BTS is not open.

"*Check that the ETs are up*."

The users have to find the ET[b]-numbers first from the link data and then check the status by MML by using those numbers. The ETs are not up, so the user takes care of that by issuing a few MML-commands. The younger user does not remember the command, which is used to open the BTS. The other user tells him to use the MML help-function. The information is found and the users get the BTS open. Now they have to call the OMC-support to tell them how to modify the TRX-numbers. In the original duo-BTS, the TRXs were numbered 1,2,5,6 and after the modification they should be consecutive numbers in both BTSs. Next, the users take a backup copy of the TRX-data and destroy the transmission links with MML-commands. There is a graphical NMS-tool for handling the links, but the users think that it is awkward to use, since the links to be destroyed must be found from a long list (100…200 links) without any search functions. **The tool has high random complexity.** Now they recreate the links by MML-commands.

The users get the BTS prepared, but they do not know the situation at the site, and hence have to call the field man. **This is interdependency of tasks. Proper system structure would make it unnecessary.** The field men are loading the HW-data and are not ready yet. After a while, the field men call from the site that they are ready. The users connect the HW-data to the BCF by MML-commands and check that the BTS is OK. There seem to be no alarms and there are ongoing calls through the new BTS, so the BTS is working. The users are surprised that it went so smoothly and inform the field men about the successful commissioning.

---

[a] The department in the operator's organisation to seek help from, when the wanted functionality is either missing or is not usable in the suppliers' systems

[b] Exchange Terminal, unit on the MSC-side of the switching/transmission interface.

They also have to take up the other BTS. The field men loaded the HW-data for both BTSs already, so they do not have to wait for that. The users connect the HW-data to the BCF. The other BTS does not open immediately, so the users give a reset and open it again. This time it works. Finally, they check the alarms and traffic through the BTS, as well as inform the field men about the successful operation. The users mark the performed operation in the commissioning list in the equipment information system. The user makes a comment about simultaneous tasks with many different groups, which must be accomplished with several different applications. This is quite common, as many people call them asking to perform various little things all the time.

Assessment

The described cell splitting is a difficult task, because the BTS must first be destroyed completely, which means that the parameters and neighbour cell definitions also vanish. Therefore, the NMS users must contact the *OMC support group* so that the aforementioned data can be stored. After this, the old elements are destroyed and recreated. If the NMS-users forget to store the cell data, they have to call the *radio network planning* engineers and ask for the parameters and neighbour definitions, which are then manually entered again. All this means a lot of interdependency and a large amount of manual effort for each of the partaking groups. Some tasks also require synchronisation of the work efforts by several groups.

The graphical tools are not very effective due to the fact that they are not very task oriented, they handle small parts of the work and possess rather low level of automation.

The field personnel often telephone the control room, as they need information about the alarms of the network elements, or if some actions need to be taken, such as locking or unlocking the network elements. The NMS-people have many simultaneous tasks, since they have to help the other groups in their tasks. That makes the work fragmentary, as they have to frequently interrupt their work and concentrate on something else for a while. The interdependence also adds the complexity of the work.

Suggestions for improvements

Once again, the task involves a lot of low level manual effort by several groups, and thus reduction of user-observable complexity and interdependence (K-factor) of tasks would be very useful. Both of these could be achieved by implementing a metasystem transition in the task. The manual data storing and destroying should be eliminated, when designing a proper tool for this task.

A good way to handle this task could be as follows. On a geographical map, the user would select 'Split BTS'-option from the BTS-menu, which would automatically create a new BCF and connect the new OMU-link. The links would be created by the transmission planner in the connection design phase, so that they are already in the database. The program would also automatically copy the BTS-data under the new BCF, as well as create a bi-directional neighbour definition between the new and the old cell and also copy the other neighbour definitions. The user could move TRXs to the new BTS or define the number of new TRXs, when the system would create them and copy parameters from the network plan. Finally, the user would send the data to the network and the view would be automatically updated.

6.8.6 Case 6: BSS-splitting

Preparations:

1. Create and open LAPD-links in the new BSC (Split tool)
2. Create BCFs, BTSs, TRXs and neighbours (Split tool)
3. Create views (Split tool)
4. Correct views (Another NMS tool)
5. Connect and activate the HW-data and BTS-SW-data (MML)

The BSS splitting:

1. Lift up possible new ETs in the new BSC (MML)
2. Lock the BCFs, BTSs and TRXs (Split tool)
3. Ask transmission equipment man to connect transmission
4. Lock, destroy, recreate and unlock the BTSs in the new MSC (Split tool)
5. Create neighbours and update CI, LAC (Split tool)
6. Unlock the BCFs, BTSs and TRXs in the new BSC (Split tool)
7. Check that the BTSs are working (MML)
8. If there are more BTSs, go to 2.
9. Remove the released ETs from the old BSC (MML)

Approximately 1 week later:

1. Destroy old BCFs, BTSs and TRXs from the old BSC (Split tool)
2. Destroy old LAPD-links from the old BSC (Other NMS tool)

3. Destroy old views (Other NMS tool)
4. Update views (Own_application_X)

The BSS-splitting becomes necessary latest when no more BTSs can be inserted under the BSC (see Figure 36). The reason can also be that the radio network is adjusted to be more logical, so that the structure is easier to understand. *This is random complexity reduction.* Then, a new BSC is added and some of the BTSs are transferred to the new BSC. It is also possible that some BTSs are moved to another, already existing BSC.

There is an application for BSS-splitting in Hi-Tech Inc.'s NMS. According to the users, it has proven to be quite useful, even though it can only be used for part of the work. The preparation work, such as filling in the list of transferred BTSs, must still be done manually. The users fill in a table of the BTSs to be switched over. After filling in the table, it is not possible to change it any more. This means that if the order of the BTSs was wrong, the work must be started again from the beginning.

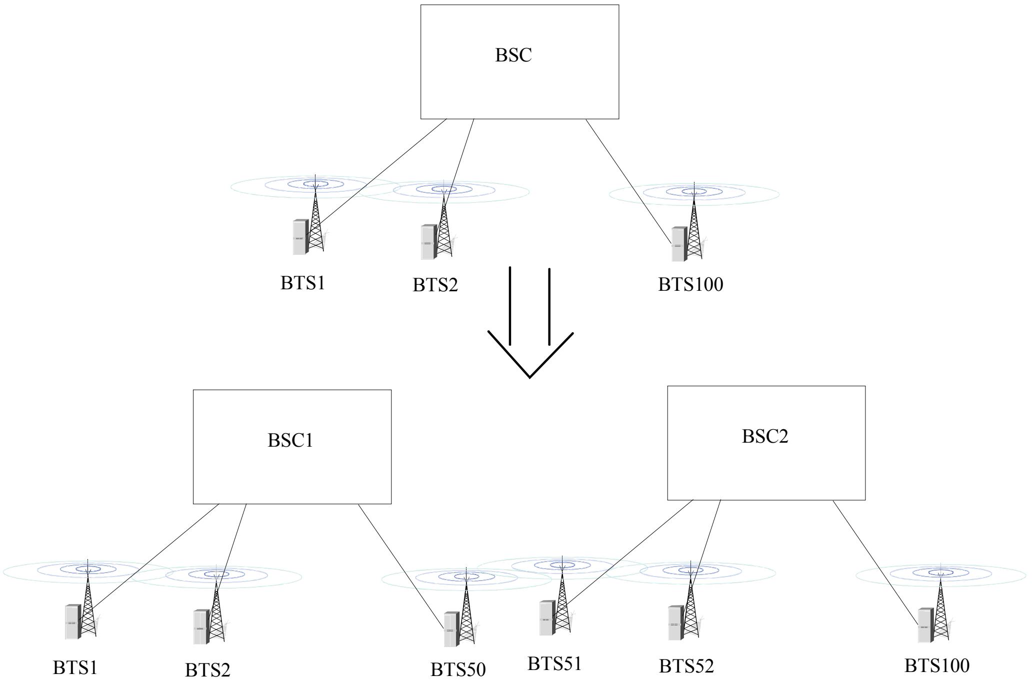


Figure 36. BSS splitting. A new BSC is added, and some base stations are rehosted to the new BSC.

Since the users think that the execution time is very long, they also divide the whole set of BTSs into several smaller sets, which fragmentises the work. In addition, the tool cannot be used if the sites must be transferred across OMC-boundaries. There is also some manual work. For example, the tool cannot correctly create the BTS-views, so the user must manually correct them. Also some of the data is not copied in the automatic process. The users must manually by MML connect and activate the HW-data and BTS software data on the other side. Anyway, the communication need between different groups (*K-factor*) and also the amount of effort has reduced a lot after taking this tool into use. Earlier, all the work was totally manual and the OMC-support had to create massive macros for neighbour setting.

Assessment

The execution times are quite long, and usually the users divide the set of moved BTSs into smaller sets. This undermines the efficiency of this 'mass-rehosting' tool. The tool does not create the views correctly, and hence the users must correct them by using another tool. In addition, the HW-data and BTS-software data is not copied automatically, which means that the users have to do it manually. However, this application is relatively good. The main reason is that the application has been designed for the operator task, not just for some hopefully useful functionality. In addition, there is a *relatively* high level of automation in the tool. The users would like to see more this kind of functionality.

The success of this tool actually proves the superiority of the task-oriented, automated applications. However, the automation level is still too low. The reason why this tool exists, is that the operator specified the required functionality and asked the supplier to create a tool for the mentioned functionality. This also proves (once again) that the suppliers can create usable tools only in co-operation with the users.

Suggestions for improvements

Although the tool is considered to be good, it is not nearly perfect. Since the task still requires many manual procedures, the implementation of this metasystem transition should be improved by extending the automation.

Some of the preparation work before the actual splitting could be automated, as well as some manual functions during the splitting procedure. Even with this application, the task-oriented tool design could be taken a little bit further. Since the OMC border problem is also limiting the use of this tool, communication between the OMCs would be required to solve the problem. The tool would require better editing possibilities, for it is not possible to change the table after filling in.

The users think that the possibility to keep a backup copy in the tool is useful in case they have to return to the old configuration. It would be even better if this was developed to be an UNDO-function. The users would like the same kind of UNDO-functions in other applications too.

6.8.7 Case 7: Switchover

The switchover procedure:

1. Lock the BTS on Supplier Y's side (NMS-user)
2. Move the PCM cable to the new BTS (field group)
3. Reroute the OMU-link to the new BTS (NMS-user)
4. Drop the BTS from Supplier Y's MSC-database (NMS-user)
5. Change the new cells to foreign cells (NMS-user)
6. Change the BSIC[a] and LAC numbers of the new cells (NMS-user)
7. Reroute the logical TRX transmission links and update BSC-database (NMS-user)
8. Unlock the BTS on the Supplier X's side and verify working (NMS-user)
9. Set the cell broadcast messages and connect the HW-data (NMS-user)
10. Clear away the now useless foreign cell definitions (NMS-user)

A switchover in this context means that the BSS of Supplier Y is replaced by the BSS of Hi-Tech Inc. In other words, the base stations and base station controllers of one manufacturer are replaced with the equipment of another manufacturer (see Figure 37). Mainly the operator's own tools (applications created by the OMC-support) are used in the process, but in addition some NMS-applications of Hi-Tech Inc. are also utilised. The BSC-views are generated from the imported BTS-data by using the Own_application_8. The tool was created for the purpose, because in this process the parameters from Supplier Y's system must be transferred into Hi-Tech Inc.'s system via a conversion. The different systems do not even have all the respective parameters. Before the actual switchover, the necessary preparations have already been made, e.g. by creating the required network elements in Hi-Tech Inc.'s NMS.

The procedure goes as follows: The field man telephones the control room telling that he is starting the work at the BTS-site and asks the NMS-user to lock the BTS on Supplier Y's side. After the BTS is locked, the *transmission equipment group* detaches the PCM-cable[b] from Supplier Y's BTS and attaches it to Hi-Tech Inc.'s BTS. There is also someone at the BSC-site to change the other end of the cable to Hi-Tech Inc.'s BSC. *There is a lot of interdependence in the tasks. The acts must also be synchronised.* Next, they call the control room to tell that the time slots[c] of the transmission

---

[a] Base Station Identity Code.

[b] Pulse Code Modulation. This is a time-divided transmission technique.

[c] Each time slot constitutes a telephone line in a time-divided telecommunication system.

line have been physically changed to the new BTS. The NMS-users reroute the OMU-link[a] by issuing several MML-commands and also verify the working of the OMU-link by MML-commands.

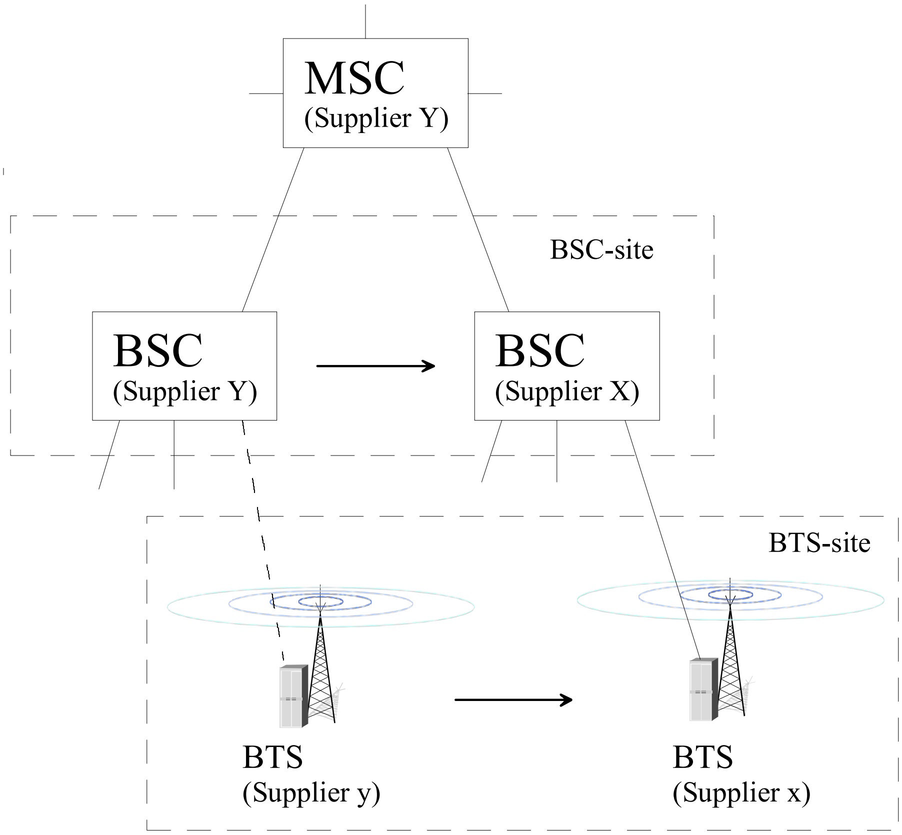


Figure 37. A switchover. Replacing Supplier Y's BSS with Hi-Tech Inc.'s BSS.

Next, the NMS-users run the change script to drop the BTS from the MSC in Supplier Y's system and to make the MSC recognise the new BSC and the new BTS. Thanks to this script, the BTS is ready for use immediately when it is lifted up on Hi-Tech Inc.'s side. All the required scripts are in text files on the hard disk. The scripts are run in the network control room by Own_application_2 + some extra MML-commands for all the new network elements to change the cells to foreign cells (belonging to another OMC). This must be done, since the systems of different suppliers are under different OMCs and if the BTSs are moved one by one, it requires foreign neighbour cell definitions. At the same time, the BSIC-numbers and LAC-numbers in Supplier Y's MSC must be changed to correspond to Hi-Tech Inc.'s BTS numbering space. In addition, the logical transmission links are changed from the old BTS to the new one (Hi-Tech Inc.'s tool) and the BSC-database is updated by using Hi-Tech Inc.'s NMS tools. These changes are only possible if the OMU-link is

---

[a] Operation and Maintenance Unit, the OMU-link is used to control the BTS from the control room.

already working. **There is a lot of low level manual effort, which could be eliminated by a metasystem transition.**

By using NMS, the users open the BTS on Hi-Tech Inc.'s side, check that it is functioning properly and that there are no alarms. Traffic through the new BTS is checked by using MML. The field man at the site performs some test calls. As a final touch, the NMS-users set the cell broadcast messages[a] and connect the HW-data with MML-commands. These are not possible to perform with the NMS. The NMS is used for alarm and traffic surveillance during the switchovers, as well as for configuring possible frequency hopping[b]. At the end of the switchover process, the NMS is used for clearing away the now useless foreign cell definitions.

The *OMC-support group* supplies the required macros for the task. The users tell that similar macros are used when a BSC is transferred from one MSC to another. According to the users, it would be much better if they did not have to call the OMC-support for things like this. The OMC-support is heavily loaded, since the control room personnel requests for help quite frequently.

"*The reason why the transfer tools do not exist, is that the supplier has not understood…has started from the idea that when you've done it once, it's gonna be like that 'till the end of time. The practice is entirely different.*"

This tells about the dynamic landscape of network operating. The network is changed all the time.

Assessment

Once again, the task requires synchronising the work efforts of the NMS-users and field personnel. This means interdependence in the tasks of the various working groups. When doing the switchover, all the logical network elements must be destroyed and re-created, which requires a lot of manual effort. As no automated tools exist, the work is done at very low level.

Since Hi-Tech Inc. has not produced suitable functionality for the task, it is done with operator's own scripts directly in the databases of the MSC and the BSC. Sometimes, it has also happened that the OMC-support has not stored the MSC-scripts to the correct place and the NMS-users have not been able to use them. In that case, the required MSC-changes must be done command by command.

---

[a] The BTSs are continuously broadcasting certain information on Cell Broadcast Channel.

[b] This is a technique to reduce the effects of fast fading and even up the interference.

Operators have to modify the network all the time. This is one example of the dynamic landscape. It also seems that the coevolutionary links between the operator and supplier have not been strong enough, as the supplier's designers have not understood the true nature of the operating tasks. It is obvious that the operator process research and feature research of the supplier have been too light.

Suggestions for improvements

As there is a high interdependency between the tasks of the NMS users and the field men, it would useful to reduce it. Since the field men are required for detaching and attaching the cables at the sites, the interdependency could perhaps be eliminated by triggering some automatic procedures when the cable is attached to the new BTS. Actually, the whole procedure could be automatically performed triggered by the cable attaching. This would definitely represent a metasystem transition.

Another possibility would be a semiautomatic procedure by using a switchover tool. Even this method would be useful, since this operation presently requires a massive amount of effort, which is further accented in the sizeable networks of large countries. The starting point for the design could be the BSS split tool, but it should be further enhanced. This could be done by more comprehensive automation, for example, by extending the automation also to the preparation work. The links could be automatically created by utilising the planned link information and also the BCF and LAC data could be automatically copied from network planning data.

In addition, the coevolutionary links should be improved to guide the tool development of the supplier to a more user oriented direction. However, to make the best of the coevolution, the supplier should have a more flexible product development process. The underlying problems and ways to solve them are discussed in chapters 7 'System usability as revealed in the empirical study' and 8 'Using the theoretical framework to find guidelines'.

6.8.8 Case 8: Fault management

The top-level user interface (TLUI) of the NMS is used for following the alarms of the network elements. The TLUI does not show an image of the physical BTS configuration, but instead a logical representation of the base station. The fault codes are displayed in hex-code, and then there is a separate on-line alarm manual, which the user must check in order to know what the code means. *This is user-observable complexity.* According to the users, the explanations in the manual are not detailed enough and they would also prefer getting the explanations directly without consulting the manual. Fault finding takes a lot of time and effort.

When the user comes to work in the morning, he opens up a BSC-view in the graphical tool TLUI of the NMS and checks whether there are any alarms blinking in the BTSs. If there are, he then starts a closer analysis. The user demonstrates some typical actions.

He opens up an MML-session from the BTS-view to study the situation. According to the user, MML is convenient for checking the status of TRXs, as well as amount and nature of traffic. To his knowledge, this kind of information is not available in the graphical tools. With the MML-commands, he checks the traffic and some other data to see how to solve the problem. This time, the alarm is of the kind that it can be acknowledged without any further action. He picks another alarm and checks the details. This one requires user intervention, and he decides to give a reset to the element. In order to do that, the element must be locked first. The user chooses to lock the BTS with a forced handover. This means that all the calls on that element are handed over to the neighbouring base stations, so that they are not interrupted for the procedure. After locking, he reopens the BTS, which also causes a reset. Then he checks the situation with MML-commands to verify that the TRXs are in loading-status. This status usually lasts 1…2 minutes, and then the BTS is ready for use again. The indicator light in the TLUI-view also goes out when the element is up and running.

There are huge amounts of alarms (tens of thousands) coming in all the time. This requires a lot of effort for investigating and acknowledging the alarms. **Concentrated manual handling of low level functionality is used instead of autonomous local control.** The serious alarms demand for further action and are therefore forwarded to the field personnel by using an in-house tool. When sending the alarms, the data must be entered manually. The user selects the receiving person, encloses the BTS identification, alarm message, fault description he displayed with MML and possibly some additional information. The receiver acknowledges the reception, after which the tool does not try to send it anymore.

Due to the last SW-update in the BSC, the operator has to take down some BTSs and recreate the links to get rid of a certain fault. The fault affects a certain signalling channel and thus impedes establishing new calls. The user starts to fix the aforementioned problem. First, he locks the BTS and then destroys it from the network, as well as from the NMS-database by an NMS-tool. The tool automatically saved the BTS under the name of LASTOMC. He verifies the destruction by using NMS. Next, he destroys the old link and recreates it by MML. Finally, he sends the backup copy of the BTS to the network by the NMS-tool and checks that the BTS gets up again. Due to the fault, the BTS must be out of operation during this procedure for about 5…10 minutes.

The measuring applications of the NMS are not very useful to the operator. They have designed many tools of their own for measuring purposes. They are using a traffic measurement tool (Own_application_9), in case they want to follow the performance of a certain BTS. The tool stores the traffic level counters for a certain period. In network management, the users usually need measurement data from a relatively short periods, for example three days. Radio network planners use longer periods, since they also have to recognise trends in the traffic. They also use Own_application_10, which utilises the counters of Hi-Tech Inc.'s NMS.

The problems coming from the end customers (telephone users) are also considered. The problem descriptions from the technical customer service are fed to a Lotus Notes –application designed by the operator. All the control room employees follow these faults based on their personal areas of responsibility. The faults that have been taken under work are marked in the database, so that the same problems are not handled by many people. There is no connection between the tool and the NMS fault management system.

Assessment

Fault localisation is not very easy, because the views of the network elements are logical, not physical to show the actual HW-racks and indicating the faulty cards. In addition, the fault codes are displayed in HEX-code and the users have to use a separate manual to find the fault description. This adds the user-observable complexity of the work. According to the users, even these fault descriptions are not informative enough to point the fault to the defective card. MML-commands are used for getting more information, which is not available in the graphical tool.

Again, the functionality has been implemented with a low level of automation. Many times, the users have to destroy the old (logical) elements and recreate them in the databases with rather large manual effort. This kind of measures are frequently required, and hence removing the manual effort would have a significant effect.

Measurement applications are not very practical for the operator tasks, and the operator has created their own tools for the purpose. The end user (telephone user) problems are kept in a separate database without connection to the network management system, thus losing the synergies between the fault diagnostics and end user problems. Combining these data, and perhaps network planning data, could enable a useful metasystem transition.

The base stations relay all the low-level fault indications to the network management centre (against the complex system principle in nature: Complexity is autonomously handled at each appropriate, hierarchical level. Refer to chapter 3.1.1 'Hierarchy, modularity and autonomy"). This means that in a large network, the NMS is flooded with tens of thousands of fault indications every day. Checking every fault manually is not possible even with the network sizes of today.

Suggestions for improvements

The system design by the network suppliers should apply the hierarchical organisation, which is characteristic of natural systems. In nature, each subunit of the organism (or system) has a rather high level of autonomy. By handling most of the problems at the level of concern, only very few problems cause actions at upper levels. Actually, the newest base stations at least partly apply this principle – they do not send all the raw alarms to the NMS, but perform some diagnostics first on their own. This is highly advisable. Task oriented design and hiding the complexity by using more autonomous lower layers (see chapter 8.1.2 'Metasystem transitions') would reduce the amount of user-observable complexity and manual effort. In addition, more informative fault messages and visual images displaying the physical BTS with fault indicators would improve the fault management. The users also think that the alarm monitoring system should have another mode, some kind of HW-mode, which would show the physical construction pointing out the faulty component, e.g. with a blinking indicator.

6.8.9 Summary of interpretations (Operator A)

According to the users, the large number of applications designed in-house indicates that either the network suppliers have not produced the required tools at all, or many of them are too complicated and slow to use. This tells about too light coevolutionary links between the suppliers and operators. With stronger links between the users and designers, the benefits of this coevolutionary relationship could be enhanced. Working for some time with the network or at least observing the real operations (see chapter 8.3.1 'Empathic feature research') would be useful for the software engineers producing the applications. Having the software people working far more closely with the users, would be the only way to improve the practicality of the systems.

"It seems that many of the applications are designed for the engineers that develop them, not for the users that will actually use them on a daily basis."

Low level of automation and a lot of low level manual effort seem to be characteristic of the system. This environment would be suitable ground for metasystem transitions (see chapter 8.1.2). As most applications have not been specifically designed for the operator tasks, the system usability is not very good. Quite obviously, the operators are also facing a very fragmented world of operating tools. It seems that no vendor offers such a comprehensive system, which allows an efficient way of performing all the required tasks without a large number of external components. If such a superior system existed, the operators would probably be using it. It is another question how easy it would actually be to design such a system, given the situation where practically all operators have a little different ways of working. Therefore, one very important starting point for the system usability is operator task research and designing efficient, best practices for handling the tasks.

The operations presently require dense communication between the various working groups. This is a sign of interdependence (high K-factor) in the operating tasks. Many times, the root of this interdependence is the system, which does not enable highly independent tasks. The field men frequently telephone the control room when they need some information, or some actions to be taken, such as locking or unlocking the base stations[*]. The NMS-users quite often need the help from the OMC-support to assist in the weekly routines, for example switchovers and rehosting of the elements. Most tasks require contribution from many different groups and the work is done sequentially in turns. One group must inform another group and ask them to do certain actions, before they can continue with their own work. This also requires synchronisation of the efforts of different parties.

Operators also have many different information systems and databases between which they have to transfer data. Many times, the transfer is done by manually entering the data again to another system. This increases the effort and number of errors.

6.9 Operator B

As the data for Operators B, C and D were acquired only by interviews due to access problems, as opposed to observations and interviews for Operator A, the format of the analysis also differs. Since we did not observe any operator tasks, there are no cases to analyse. Instead, we have the characterisations of the operating situation and technology as the interviewees perceive it.

Operator B has a medium operating experience and they are not willing to invest heavily in their own technical expertise, but rather outsource it when necessary. They have a large network.

---

[*] This was also noted in the research by Lucas/Meech/Purcell 1997[166]. They reported that many users work in conditions, where constant telephone interruptions are common and multiple systems are used.

6.9.1 Configuration management

According to the interviewee, the operator is struggling with configuration management. Information on the existing network elements, e.g. amount of TRXs, number of cabinets, element arrangements etc. of a given area or site, should be in the NMS as a some kind of hardware inventory system. It would be useful to have a map showing all the sites in the area and what sort of hardware there is, as well as what features are on. In addition, reporting of this information should be efficient, for example in the form of automatic hardware changes tracking. **This indicates that the operating landscape is dynamic.**

"We need the ability to see what is actually out there, because the systems we have currently tend to tell you what is planned rather than what is actually there."

The interviewee feels that it would be better to see the geographical layout of the network rather than the logical one. This would help in dealing with e.g. adjacency problems, as it is much easier to see where the problems lie when the sites are grouped geographically. The operator has started to convert the information presented in the GUI into geographical form, i.e. a map displaying the sites on their geographical locations[a].

6.9.2 Fault management

The operator actually has teams for developing the configuration and fault management in order to improve the system usability of the network management systems. **This hints that the supplier's product development is too slow for the operating landscape, for the operator itself is developing the system.** The interviewee is trying to find ways to improve the network management system so that less alarms would go up to the NMS-users and so that the displayed alarms would be presented in the right format. There is no product available from Hi-Tech Inc., that could handle the equipment of numerous different vendors, so the operator bought a third party tool, which was able to do it. This front-end monitoring system, which interfaces with Hi-Tech Inc.'s system, is used for monitoring the whole network consisting of equipment from many vendors. In other words, Hi-Tech Inc.'s system is used as a network element managing system attached to the front-end, which

---

[a] One could argue that things like this are about functionality, not about usability. However, the system usability is defined with the aid of the effort users must invest in operating the system, and this action can be understood as reducing the effort of viewing a large network. The operator might have to invest a lot of effort by using external tools for finding out the geographical relations of the network elements in everyday operations. With the new system, this is no longer necessary, as the required information is readily available.

combines the systems of different vendors and provides a network wide view (refer to Figure 27). **The operator is actually applying metasystem transitions here, as they introduce the network management level on top of Hi-Tech Inc.'s element management.**

The alarms coming from Hi-Tech Inc.'s system are sent to the front-end system, where they can be correlated and combined from different vendor sets. The huge number of the alarms (tens of thousands per day) is a problem, since it is very difficult to make sense out of them all. Hi-Tech Inc.'s system has always generated more alarms than the other vendors' systems. One also gets several alarms related to the same fault. If the system were clever enough, it would only show the most relevant one. **This shows very low level of automation and lack of autonomy in the subunits.** The information in the fault descriptions is not detailed enough. Sometimes, this leads to a situation where the field men go to the site without the right spare parts, and hence cannot solve the problems there. Therefore, it would be good if they knew beforehand exactly what the problem is. The NMS-users do not have enough time or training to narrow down the problem precisely enough. It would be an improvement, if they could e.g. see a graphical representation of the BTS rack with the faulty card indicated. The interviewee has used other tools with such a level of detail.

The operator is trying to achieve a stage called front-end automation in the fault management. They have asked Hi-Tech Inc. to investigate the possibility of combining fault management information with performance management information on the network element layer, i.e. to have the system bring only the most severe service affecting faults to the top level and automate as much as possible at the element management level. **Here the operator has tried to get Hi-Tech Inc. to follow the metasystem transitions principle. Also the hierarchical organisation and autonomous subunits used in nature is visible here (refer to chapter 3.1.1 'Hierarchy, modularity and autonomy').**

There are also error messages in the NMS itself, that need to be laboriously individually cancelled. The messages give a hexadecimal code signifying the process where the error occurred. 90 % of the cases it is difficult to figure out where the problem really is. There is a special staff for taking care of the NMS and dealing with the problems users are experiencing in using the applications. If they cannot figure out where the problem lies, they have to contact Hi-Tech Inc. in their home country. If the problem occurs in an application, which is used actively, it very much affects the overall system usability.

"Customers do not want to know what the problem is – they only wish to see the system working."

6.9.3 Performance management

There seems to be a gap between the information acquired with field-testing and the information gathered by the system itself. The interviewee explains that the operator is able to pick up faults when they happen, but they are not good at looking at degradation in quality over time. He would like to be able to trend things, but it is difficult to put together a profile of the network with the current system and tools. The use of measurement tools is too black-and-white. There could be 20 cells just hovering about a certain threshold, which constitutes a bigger problem in terms of quality than having one cell black out completely. One needs a system, which tells when there has been a change in the network statistics and how significant. It should also track combinations of changes. *This indicates low level of automation and dynamics of the landscape.* This kind of information would be valuable for radio network planners. The interviewee tells that the way to make better use of the information coming up from the network is to integrate all the systems by linking the planning, performance and data build tools. After this, one could start doing the 'clever stuff' like automation. *This insight points toward metasystem transitions.* The interviewee has been investigating which performance measures are needed to develop self-tuning and self-healing features. *This fits the description of complex adaptive systems.* Only a tiny fraction of the existing measurements are needed for this.

According to the interviewee, the current performance management is poor, since one can only handle measurements individually. This is very time consuming and frustrating. As an example, he tells that when they are moving NMSs to a new site, they have to manually recreate the measurements into the new system. This means loading the measurements one-by-one from every BSC and individually creating them in the OMC-database. They have created a script to switch the measurements on or off and another one to synchronise the OMC-database with the NMS. The management of the measurements and statistics would require improvement, possibly a browser based application equipped with a GUI. It might also be useful to give other groups, for example the field men, access to the OMC information. Then the field men could figure out themselves what the problems are with the site, and do some simple operations, such as locking and unlocking the BTS instead of having to call the NMS-users. *This is reduction of interdependence (K-factor).* This would save a lot of time and effort. In addition, the number of keystrokes required to lock a BTS should be decreased, since presently it requires locking each TRX individually and then the BCF.

The lack of processing power and data storage capacity on the OMC is also bothering operators with large networks. When doing trials, a lot of statistics need to be studied. The users can only turn the counters on for a short period to gather the statistics because of the limitations posed by the systems on which the OMC is running. There are also limitations as to what information can be extracted from the OMC. The number of counters is not the problem, but the lack of knowledge on what counters exist and where. **The work is done on the level, where individual counters are followed. E.G. complex adaptive system technology would allow a metasystem transition and elimination of the manual use of counters.**

The operator also uses some third party products to measure performance of the network, for example Metrica and TEMS. They also use three or four tools to analyse the data gathered with the drive-around tests. One of the tools is in-house developed.

Another interviewee tells that the PM (Performance Management) tools have been very confusing to work with in his opinion. As more and more sites are added into the network, the operator must start removing some of the BTS adjacency definitions. It would be nice to see a tool, which would graphically display the usage of adjacencies between sites. The little-used adjacencies could then be removed to optimise the number of handovers. If the system were clever enough, it would automatically remove the seldom used neighbours. **This kind of task is suitable for complex adaptive systems.** Currently, the interviewee can analyse the usage of the adjacencies with offline software packages, but he still has to do the changes manually in the NMS. The functionality should therefore be in the NMS. In its current state, the adjacency management tool in the NMS is purely for *creating* adjacencies, there is no real management involved.

From the adjacency management point of view, the worst problem is the inability to see the adjacency information across the OMC-borders and keep the adjacencies up-to-date. This is not good from the data integrity point of view, as errors can easily occur. One of the common problems is that after a frequency plan change, someone forgets to update the frequencies of the adjacent cells in the neighbouring OMC. This causes handover failures and an increase in the number of alarms.

"The OMC boundary management is in shambles."

Another thing the interviewee has problems with is tracking the changes made to the numerous parameters and the effects of those changes. **This indicates dynamic landscape. A metasystem transition by using e.g. complex adaptive systems would allow elimination of the manual tracking.** The only way to keep track of the customised parameter sets, is by keeping notes outside the NMS. The interviewee hardly ever changes half of the parameters due to the problems with parameter management. If there was an efficient parameter management system being able to cope with the number of parameters, he could change more of them. **This is one of the reasons why the networks**

are never in optimum. The operator has created in-house performance management tools far superior to those that Hi-Tech Inc. offers, claims the interviewee. Another interviewee also tells that most of the parameters are not changed. In his opinion, some of these parameters could be removed.

The interviewee tells about the laborious parameter changing procedure. If he would like to, for example, turn on the frequency hopping across a whole BSC via using the File Export/Import, it would be a very effort intensive process. First, he would have to export the data of the whole BSC. Then, he would have to apply scripts to make the changes into the file and import it back. Given the amount of data, the margin for errors is quite large. If there were an option in the file export to only export certain parameters, the savings in time would be significant. The ability to move across only specific parameters would not only save time, but also put less load on the network. In some places of Hi-Tech Inc.'s network management system, common sense is lacking.

6.9.4 Service management

It is not merely the radio network, which needs improvement. The operator needs to have a better view of the services as well (refer to Figure 27). The basic quality of the network will always be maintained, but the services will become the bread and butter of the operator. The effort put into managing a service depends on its nature and how many customers are tied into it. Of course, the system itself should enable straightforward service management. The infrastructure has to be in place first, so that the new service can be plugged into it to allow immediate visibility on the performance of the service, the number of users and the profit. This kind of ability does not exist yet. The interviewee thinks that currently most of the operators cannot see how the end-to-end services are behaving and what kind of effect failures have on the services. A prerequisite is that the network can be made easy to manage, so that the attention can be switched to other areas. **Realising the service management layer is a metasystem transition.**

"If you know what your configuration is, you know how the network is performing and you know what faults you've got, you've got a far better picture of the overall service you can deliver."

Another aspect of the system is that the information currently available is very technical. It assumes that the person accessing the data knows a lot about the equipment in the network. The operator would like to give the customer care people visibility into the status of the network, along with the technical view. It would also be important to get information, which could be used by e.g. marketing and ultimately allow the customers to view the information.

6.9.5 User-observable complexity

The interviewee states that the operator personnel is not experienced enough to handle the work. *This could also be interpreted the way that using the system is so complicated that it requires a long experience of using the system.*

The operator uses an optional feature from Hi-Tech Inc., let us call it the Complex_Feature. The supplier had to arrange specific courses for the operator to explain how the feature actually works. The operator also had a rather complicated trial to test the mentioned feature a few years ago. Nowadays, the feature is in use in a certain area, yet the people responsible for maintaining the feature do not fully understand it due to turnovers in the staff. *This is a good example of user-observable complexity.* There are many parameters related to the Complex_Feature, yet the operator is using the same sets for long periods of time without really looking into them. *This means that the feature is not working optimally.* The interviewee believes that the operator is not benefiting at all from using the feature, rather the opposite. The operator is considering to develop some kind of intelligence to automate some definitions in the Complex_Feature, so that it would be less complicated to use. *This is reduction of user-observable complexity. More precisely, it is reduction of random complexity – effective complexity is probably increasing, as the automation comes on top of the original feature.*

In the interviewee's opinion, so many parameters should not be available, since no-one plays with them. Most of the parameters could be self-tuning, as the data for computing the values are already available. *This is a good application area for complex adaptive systems.* Some of the parameters could be totally removed even without major changes in the algorithms. *This is reduction of user-observable complexity.*

Ultimately, the operator wants to have a some kind of adaptive system, which could interpret the measurements itself and tell what the problem is, and even fix it. It is getting too labour intensive to do the analysis manually. The interviewee is certain that some high level things get missed, because people are too busy with the low level details and the volume of things they have to look at. Consequently, it would be good if the technology would take care of the low level operations, letting the people take care of the high level. *The operator is actually suggesting a major metasystem transition.*

There are too many applications in the NMS, like the BCF-software related ones, for just one operation. There are also some non-network management related applications due to historical reasons. For example, applications like User Group Profiles or Authority Manager are so closely related that they should be bundled into one application (e.g. NMS User Manager). *This is reduction of user-observable complexity. More precisely, it would be reduction of random complexity, if the effective complexity is not changed.*

The functionality in the NMS to re-host BTSs and BSCs does not work the way the operator does the operations. They usually want to re-host 50…100 cells, which requires a lot of time and effort for preparation work (manual preparation). One simply cannot re-host such an amount of cells with the NMS and do it as quickly as the operator wants it to be done. There is an application called BSC Split for this purpose, but it does not transfer the data fast enough. It also works only within one OMC at a time. For the mentioned reasons, the operator does the re-hosting very much manually. A network plan is used to build the site data with NMS tools, after which the adjacency information is inserted by running some scripts. There are plans to develop the system so that it could automatically build (logically) the network from the plans. *This is a metasystem transition.* The business demands that the process of building the network be made more automatic on the configuration side. The biggest things that need to be done are the integration between the configuration and fault management tools, as well as integration between some other tools because of the multivendor nature of the network. *This also points toward a metasystem transition.*

6.9.6 Product development

One of the most annoying things, according to the interviewee, is the lack of synchronisation in the BSS and NMS software levels. For example, at the moment the operator would like to implement a new optional feature from Hi-Tech Inc. over 10 000 sites, but it will be very hard to do as the feature is not yet supported by the NMS. Some of the work is done with NMS and the rest with macro commands directly into the BSS. The supplier is getting a lot of inquiries from the operator as to why things have to be in a certain way, for example to fully support all the BSS parameters in a certain release, they have to have a certain, much later version of the NMS.

Many times, the operator personnel perceive the manuals to be out of date, although it is mentioned that certain features are not supported by this version of NMS. In addition to the problems that the operators have with this situation, it also takes a lot of co-ordination between the support groups in the supplier's organisation. ***This is a coevolution issue between the product lines within the supplier's organisation.***

The approach should move into managing capabilities instead of separate features. ***The interviewee is suggesting a metasystem transition.*** It is a continual job to try and keep track of the feature catalogues, what is new and what is an improvement to something implemented before. The interviewee would like to have more visibility to what will be put on the network feature roadmap and what functionality it has. The operators could then comment on it and suggest changes before implementation. Now the operator usually gets a functional description of the feature *after* it has been designed, implemented and tested. The interviewee suggests that a lot more co-operation should take place between the operators and Hi-Tech Inc. in terms of developing the software. ***The interviewee is asking for stronger coevolutionary links.***

"When you develop software, don't just get the developer to do it, get the actual users to help."

As it is also hard work reading the complicated functional descriptions, an animated visual presentation could save time and effort for the operator and also lower the threshold of getting acquainted with the descriptions. For example, the material of the Complex_Feature should be in multimedia format.

According to the interviewee, the development of the required features takes too long, as the problems arise over the period of a few months. The operator cannot afford to wait for two years to fix the problems, which is the way Hi-Tech Inc. does their business. ***The dynamic landscape is changing faster than Hi-Tech Inc.'s product development process can follow.***

6.9.7 Operating processes

In the user's experience, the NMS works in a particular way, so one learns that way and does the work accordingly to achieve his goals rather than adapting the NMS to the tasks. ***Hence the tools define the ways of working.*** According to him, the NMS should be more flexible. When the NMS tools are very unusable, the operator either uses MML-commands and command scripts or other tools of their own to do the tasks. A major difficulty is the amount of calls from the field men asking the NMS-users to do this and that. ***This is a down side of concentrated control. Not everything should be concentrated.*** This draws them away from the network monitoring function.

The interviewee and his colleagues are doing most of the site related tasks, such as resets, from the control room, since the field men cannot do them at the site with the current equipment. ***This is an example of interdependence in the tasks of different groups. It is also an example of the system stipulating the way of working.***

It would be nice to see an operator driven approach to the development of NMS. The interviewee also accepts that the suppliers would recommend best practices for the operating tasks, as long as they are based on proper practical research and are more efficient than the current processes. ***Interviewee is interested in improving the coevolution. He also accepts recommended best practices.*** He does not see how the operators could turn down the supplier's advice on how to do certain tasks, if the supplier could offer solutions that clearly solve certain major problems, for example mass rehosting of BTSs[a]. ***The example indicates that the management applications are not designed according to the operating tasks and lack efficient mass processing tools. It also tells about the dynamics of network operating.***

6.9.8 GUI vs. MML

The interviewee does the everyday work, e.g. site integration, with MML-commands, because he has better visibility this way. One gets very good feedback from the MML and using it is quick compared to NMS. On the radio network side, the NMS is getting slightly better, but there are still things one cannot do with the GUI, for example, the cell broadcast message definitions. In addition, checking the status of a BTS coming up from a reset must be done by MML, as one cannot tell the exact phase from the change of colours in the GUI. With the NMS-tools, also the delay is long. One cannot tell whether it is the OMC, BSC or BTS that is delaying things, but it is annoying. Otherwise, they can do their work by using many NMS applications at the same time, but it is complicated and does not give much feedback.

Drawing of the BTS views is time consuming. If the user could only tell the required number of TRXs and BCFs, time would be saved. Now it takes 30…40 minutes to logically build a large site. With MML, the task can be done in a much shorter time. Some sort of automation is needed in the

---

[a] This is a good example of the functionality/system usability dualism. One could say that creating a solution that solves a major problem is about functionality, not about usability. But when interpreted from system usability point of view, the new solution could dramatically reduce the manual effort, and hence increase system usability.

process. The NMS-views are getting congested as more and more sites are added, which makes it harder to find the element in the views. **This is an example of random complexity.**

"...that [using MML] has got to be quicker than fiddling around with the GUI and trying to find the cell."

The operator is not using the NMS for managing the network as much as they could. The NMS is used as an alarm viewer and MML as the tool. Even Hi-Tech Inc.'s own personnel creates the sites with MML instead of GUI when working with the operator. An NMS-user states that one reason why people are not using the GUI very much, is that they have a long experience in using the command based functions (MML) and are used to doing the work the old way. MML-users claim that MML has better feedback, whereas when using GUI one has to trust it blindly. However, they admit that there would be certain advantages in the GUI-tools, if they were designed to support the tasks properly.

"The graphical interface is there to simplify things that would take time."

In the user's opinion, the GUI is, in principle, suitable for doing large operations. However, it seems that the supplier has not completely succeeded in fulfilling this expectation yet.

6.9.9 Summary of interpretations (Operator B)

The NMS and planning systems do not contain information about the existing network equipment (hardware) and change history of parameters, hence the operator has to maintain this information elsewhere. NMS displays the network layout in logical form instead of a geographical one. Many interviewees mentioned that a geographical layout would be more useful, e.g. when dealing with cell adjacency problems.

The operator is trying to improve some things in the NMS, as the supplier seems not to be changing them. For example, the operator has been struggling with the huge number of alarms for many years. Hi-Tech Inc.'s system generates much more alarms than any other vendor's system, and the information is not detailed enough to pinpoint the faults. The operator would like to predict the failures and trend certain things, but currently there is no way to do this. It seems that the operating landscape changing speed is faster than the supplier's product development can follow. Also the alarm handling has been designed so that the network elements have no autonomy, but leave everything to the operating personnel in the network control room. This principle is against the complex system building principle of nature.

There is a lot of low level, manual effort in everything that the operator does, even in those tasks that have automated tools, since the tools have been designed to only cover small parts of the tasks. The operator still has to put a lot of effort in the preparation work before the tools can be used, and also do some corrective actions after using the tools. In other words, the automation so far has been too narrow to be really effective. The information coming from the system is very fragmented, and there are too many separate applications in the NMS, as well as many in-house developed and 3rd party tools in use. All this adds the user-observable complexity. The management system should combine various pieces of information, e.g. planning, performance and fault management, to offer more useful information and control to the users. The development should be towards capability management instead of the current management of separate features. It seems that the operators do not want to put their effort on maintaining all the little details of the system, but instead concentrate on service management (see also De Vos 1997[165]). In other words, the situation calls for a metasystem transition.

The users consider cell adjacency management more or less only adjacency creation, since there is very little means for actual management in the tool. It is not possible to see the adjacency information across OMC borders, and all adjacencies must be manually updated from the adjacent OMCs, e.g. after frequency plan changes. Parameter management takes so much effort that the users avoid changes. When new parameter sets are moved across from the planning tool, it is only possible to export the whole set of parameters, not individual parameters. In addition, there is no way of keeping track of the parameter changes within NMS. Once again, there is a lot of low level manual effort. Complex adaptive system technique would be useful here to remove the manual actions with the details.

Some features are too difficult to use, a good example is the Complex_Feature. According to the operator, they are probably not benefiting from the feature at all. With some sort of auto-configuration, also the Complex_Feature could be useful for the operator. Many interviewees also said that so many parameters should not be available, since no-one uses them. Users would like adaptive systems that can perform the low level operations autonomously. A metasystem transition is called for. As this domain is the very application area of complex adaptive systems, this technique should be investigated here.

The operator would like to give the customer care people visibility to the network status, but only technical information is available. This is not suitable for non-technical people. The operator would also need to have a better view of the services, but service management is not really supported yet. The operators cannot see how the end-to-end services are behaving. The system should allow immediate visibility on the performance, number of users and profit of the newly plugged-in service. Also here, some sort of metasystem transition would be required.

There is no synchronisation between the BSS and NMS releases, which demands extra effort from the operator. The coevolution within the supplier has not been sufficient, since they have not been able to offer the management tools together with the network features. In addition, all the solutions go through a long development, during which the operators cannot affect the product features. This means that the delay of getting the new functionality is long and the result leaves room for improvement. The underlying reason might be that the development process of the supplier is not very suitable for the dynamics of the landscape. As the operator pointed out – they cannot wait for two years for the solutions.

Many times, the tools dictate the way of working, for the network management system is very inflexible. The users would like to develop the way of performing the tasks, but the system does not allow new ways. On the other hand, developing the operating tasks could improve the efficiency of the work, but for the best results it should be combined with the system development. More operator drive in the product development would be appreciated. The interviewees also stated that the operator would accept new operating work procedures, if more efficient ones were available. This suggests that operators would be willing to improve coevolution with the suppliers to increase their fitness.

MML is still used quite a lot. The Graphical User Interface of NMS does not provide all functions that are possible with MML, and many tasks performed with the GUI are slower than with MML. Those persons, who have first learned the MML-way of doing things tend to use it even in situations that can better be handled with GUI, and those who started with GUI in the beginning tend to use it more. Many times, Hi-Tech Inc.'s own personnel teaches the use of MML to the operators in the beginning instead of the GUI-tools. In this way, the supplier teaches wrong operating models to the operators. This is an example of negative influence in coevolution. The use of a GUI should be more efficient than a low level command language, such as MML, but the current realisation does not completely fulfil the requirement yet. The set of graphical tools could use a metasystem transition.

6.10 Operator C

Operator C is a relatively inexperienced operator, which relies heavily on the network supplier's help in using the network equipment and guiding to operate the system. They have a large network.

The following material contains also information from some very experienced employees of Hi-Tech Inc that were working with operator C.

6.10.1 Radio network

The situation changes in the network based on how the subscribers move and behave in long term. The network has to be changed accordingly and this takes time and effort. The operator must also continuously put effort into maintaining the quality of the network, even when keeping it constant. *This is a good example of a dynamic landscape.* The maintenance effort is a twofold issue: 1) part of it goes into maintaining the basic platform, keeping the network available to subscribers, and 2) part of it goes into enhancing the performance, e.g. by switching the capacity according to the geographical changes on the telecommunication traffic or tuning the performance by changing the parameters. *As the dynamic landscape changes, the operator tracks the changes by manually altering the network and the parameters.* Since there is a large number of low level parameters, one can end up staring at the individual parameters, when in fact one should consider how the network behaves on a higher level. *The users are mislead by the user-observable complexity. A metasystem transition would be required to solve the problem.*

The capacity of the network must be increased based on how the number of subscribers increases. The closer to the theoretical frequency band limit the network is expanded, the more costly it is to keep the quality the same, not to mention raising it. Even if the amount of subscribers did not change, the surroundings change affecting the network. *This tells about dynamic landscape.* The tools should help the operator to form a view of the future, or at least increase the time the network remains valid. They should be capable of 'pre-emptive thinking'. *A good technique for adapting to a dynamic landscape is the complex adaptive systems. Nature has proven this through the long evolution.*

The interviewee agrees with the notion that parameterisation is difficult. For example, most of the network planners in the project do not understand how the features or the handover algorithm works. *This is user-observable complexity.* Even though Hi-Tech Inc.'s personnel responsible for the optimisation are experienced, there is always a lot of patching to do after a major network modification. Also using special features, such as the Complex_Feature, is difficult, particularly in a rapidly expanding network, since the managing takes so much effort. In a more stable network, the situation is a little easier.

Another interviewee tells that the operator has been using the Complex_Feature, although it is considered to be very difficult with all the complex parameters and frequency settings. It takes a lot of resources to maintain, yet the network quality is not very good. The interviewee states that using this kind of features is not profitable, if the operator cannot get them working properly. In the local office, even the most experienced employees still do not properly understand the complicated feature. *This is user-observable complexity.* The interviewee feels that audiovisual multimedia documentation explaining the algorithms, as well as some other complex features, would be a useful idea. However, it does not remove the underlying problem that the implementation of the actual features is too difficult. The interviewee would like to update frequency plans little by little, since updating the parameters in Hi-Tech Inc.'s system is so time consuming that the work cannot be completed in one night, which is the time frame set by the management. Other vendors' BSS is simpler and easier to use than Hi-Tech Inc.'s and similar tasks can be accomplished in another supplier's network in a few hours. *Obviously, the other system has a lower user-observable complexity. This also seems to translate to smaller effort in using.*

6.10.2 Network planning and tools

The network planning tool does not directly transfer data to the NMS, but via Excel-tables. This makes the data exchange slow and inconvenient. After all, the planning tool should be able to communicate with the existing network (via NMS) quite well in order to make things go smoothly. The NMS contains data, which would be useful in network planning, but the data is scattered around the system. The planners have to gather the data, analyse them, design the alterations and implement the changes by modifying the low level parameters. If the system were clever enough, it would direct the behaviour of the network based on higher level suggestions from the users. *The interviewee is suggesting a metasystem transition.* The interviewee thinks that there are also too many separate applications in the NMS. *This is user-observable complexity and falls into the category of random complexity.*

The interviewee dreams of a planning tool that would present the site plan, on which one could get a panorama view with all the relevant information (antennae, cable length etc). He explains that one could, for example, see the results of antenna down-tilt modifications, if the planning tool and NMS were able to properly exchange information. One could take the down-tilt of every cell and present it on the site plan, and then import the dropped call rate from NMS and print the combined data. After making the modifications, one could compare the dropped call rates and deduce whether the modifications had any effect. He tells a few example scenarios of how he would like to do the work. Although the visions could well be realised with the present technology, nothing like that presently exists. *This seems like a coevolution issue. The present coevolutionary links between the designers and users are not sufficient. Perhaps also the landscape dynamics are too fast for the development process.*

Example scenario 1: The user could check the network status in the morning and possibly notice a non-functional cell. After modifying the parameters, some network planner would go to the site with a BTS troubleshooting tool, find out the problem and then further modify the parameters and follow the status via a mobile connection to the central point.

Example scenario 2: A network planner is doing drive tests in the field. He clicks a cell in his portable system to find out the neighbouring cells in the network plan. Now, he receives the adjacent cell listing from the main system and continues collecting the measurement data. Then he compares the gathered data to the plan shown on the portable system. After comparison, the user modifies the parameters and sends the modifications to the NMS getting an updated status back to the portable tool.

Example scenario 3: (further integration of scenario 2 into operator's system) The operator's customer service marks on the network view a position where they received a fault notification. After this, the system automatically sends the information to the network planners indicating the fault. The closest network planner then drives to the area for investigation. By using the up-to-date network view, the customer service personnel informs the customers of the situation and gives an estimate of the repair time.

6.10.3 Product development

The existing optimisation database containing all the optimisation actions (e.g. solutions to the found problems) ought to be in the planning tool. Two additional databases were developed in Hi-Tech Inc.'s local office. Currently, they are working on a solution, which would combine all three databases. The applications should be able to transfer data between each other, and yet access only one central point (NMS) that is responsible for distributing the required data to the applications. All manual interfaces are obsolete. NMS and some sort of central planning database could be physically integrated allowing the users to download network planning information into their laptops to see the changes in the network and also retrieve the NMS measurements for analysis. This would enable the planners to see the network as a comprehensive package, not just a site plan and separate lines of numbers.

They have also developed planning processes for site surveys, indoor planning and parameter optimisation. The interviewee finds this local development as a problem, since the tools and processes should be globally provided. There is no global tool for managing site data, e.g. antennae, cables, digital images of the site and surroundings etc. Such a tool was locally developed as an Access-database solution and is now used in many different projects in the country. This kind of database should be in the planning tool, not as a separate program. The idea behind many of the improvements is to decrease the amount of manual effort. Yet, all the applications that are used are so-called 'quick-and-dirty' tools having nothing to do with Hi-Tech Inc.'s commercial products. All the global planning tools have been designed for 'green field planning', although this forms only about 10% of the overall planning. **This is a coevolution issue. The present coevolutionary links between the designers and users are not sufficient.** The interviewee has been criticised for not using the global tools, even though the reason simply is that these tools are unsuitable for his tasks.

"The tools should be designed for planners, not for the engineers that develop them."

The *operators* also use centralised database solutions they have developed or bought. It is not very difficult to export the NMS counter data, for example, into Access-database. However, the interviewee does not really like the ad-hoc applications, as Hi-Tech Inc. would have the resources and possibilities to develop a solution superior to those of the competitors. The interviewee does not see any problem in the current model of using key customers' needs as the basis for specifications. The real problem is that the products do not turn out to be what the customers want. **The coevolution is not working properly, since the deficient coevolutionary links are not providing enough selection information. The product has been sliding down from a fitness hill. This could constitute an error catastrophe.**

6.10.4 Operating processes

Hi-Tech Inc. has guided the operators in the country to run their networks less than efficiently. A new operator follows closely how the supplier constructs and operates the network. In the construction phase of the network, the supplier's engineers access the system with MML, and at the same time teach the operators wrong operating models using the low level command language instead of NMS. The interviewee explains that people get lack of trust for NMS after learning the direct access (MML) to the BSS. When one learns a certain way, it is not easy to start using another way requiring a totally new mindset. *This is a coevolution issue. The supplier has not developed the coevolution to the right direction. Teaching the processes correctly from the very beginning would be a mutual benefit and give coevolution better chances to improve the system usability.* Due to the late installation of the NMS, the supplier's employees cannot use the NMS in the beginning. However, the early deployment of the NMS is merely a question of timing and planning. Installing the NMS takes less time than installing an MSC, so it would be entirely possible to get the NMS working early enough. Performance issues are another factor. Some tasks are slower to accomplish by using NMS than MML.

Some operators have developed their processes and made them more efficient. After this, e.g. transmission and BTS duties have been given to the same group. In practice though, the people are not many that can manage both BTSs and BSCs, because there is so much that must be learnt. Hi-Tech Inc. has hired some people from other operators to teach the customer's personnel the operating processes. According to the interviewee, there should be a number of operating models, derived from the well working examples that could be recommended to the new operators. *This is a coevolution issue. The interviewee is suggesting closer coevolution. He is also suggesting to use recommended operating processes.*

In the future, when new technology fundamentally changes the network infrastructure, new requirements will emerge for Hi-Tech Inc.'s system. The system should support the operator's whole process of developing new services from the design stage to the implementation and release. Response time will become an important issue in terms of competition between operators. This means that the management system should be very flexible. The platform and equipment sold to the operators should well support the way they operate their network. *This is a coevolution issue. As the changing speed of the landscape seems to be increasing, the system should be more flexible but also the product development process of the supplier should enable fast tracking of the dynamic landscape.*

6.10.5 Fault management

The NMS suits better such operators who know Hi-Tech Inc.'s system down to the details and can pinpoint the problems, but on the other hand the real problems can also disappear under the massive flow of alarms and other information even with more experienced operators. *This is user-observable* *complexity.* In Hi-Tech Inc.'s system, the amount of alarm information is huge and it needs to be filtered manually. The interviewee tells that a team from Hi-Tech Inc. was implementing alarm filters to reduce the number of alarms in the network from 20000 to 11000 a day[a].

Some of the advanced alarm control tools are too complicated for the novice users, as one has to work for some time with the NMS to be able to set up all the correlations. The operator needs a more intelligent alarm handling system than the NMS presently is. *The intelligence could also reside in the network elements to avoid the chaos in the concentrated management system. Natural systems distribute the intelligence to make the subunits as autonomous as possible to avoid this kind of problems.*

Some interviewees mention the information in the alarm manual, saying that it is not detailed enough to really deal with the problems. They cannot find the problem causes easily, because the information is too vague. One of the interviewees tells how he normally acts in a case of an alarm. First, he gets some related information from the BTS by issuing some MML commands. Next, he types the alarm number into the alarm manual to obtain a description of the alarm. Then, he tries to fix it by MML, and if this is not successful, he passes the alarm information to the field men whose duty it is to repair the faults. The field men verified that the short descriptions are not very helpful in finding the faulty cards. Sometimes they cannot fix the problem at all, because they cannot find the reason for it. In that case, they call Hi-Tech Inc.'s engineering personnel, who will then try to repair the fault.

The same people in the operators organisation do also the BTS commissioning works according to a similar process than the one described earlier in chapter 6.8.4. According to the field men, it would be more convenient, if they could do the task autonomously without asking for help from the control room. They would also like to access some particular cell information, such as dropped call rate and adjacent cell information for fault finding purposes. *This would be reducing the interdependence of tasks.*

The interviewee wonders why there is no such a feature in the NMS, which would enable one to see an image of the actual equipment and the faulty unit there. By clicking the faulty unit, one could get

---

[a] One product line has designed elements generating all these alarms, and another product line has then designed filters to get rid of them. It is actually mentioned in Hi-Tech Inc.'s NMS documentation that "…one of the best ways to make monitoring more effective is to reduce the number of incoming alarms".

the related information. The first line maintenance could find the faults much more easily in this way, and the amount of effort would be reduced. The tasks would also be easier to learn.

6.10.6 Configuration management

It is not easy to load a new network plan into the network. For example, the planning tool file is first converted to Excel-format and then imported into the NMS. Every time the conversion is done, errors occur. It takes many days to check the imported configuration for errors. To circumvent this problem, the operator has developed a tool, which combines the functionality of the planning tool and the NMS at some level. *Again, the operator has developed the system, rather than waited the system supplier to do it. Fixing this kind of problems is clearly the supplier's responsibility. The landscape is more dynamic than Hi-Tech Inc.'s product development can follow.*

The interviewee finds the consistency checking tools of Hi-Tech Inc.'s system unreliable and also the foreign neighbour management on the OMC boundary area unsatisfactory. The operator uses the NMS to download data to the network and their own tool to check that the data is correct. They have started to use an Access-database application to check the adjacent cell information on the OMC boundary area and MML-scripts to correct the possible errors. The operator is also developing a tool of their own that would be used for network management instead of Hi-Tech Inc.'s NMS. The interviewee considers the NMS too difficult to use. *This is user-observable complexity.*

Another interviewee carries out field measurements for parameter optimisation purposes with the operator's own tool. After gathering the measurement data, he analyses it with the planning tool and some of the operator's own tools. Then, the parameters are changed and verified with NMS statistics. According to him, it would be very good, if the field measurement data could somehow be combined with the NMS measurements to give a better view of the situation. *This would be a metasystem transition.*

6.10.7 Summary of interpretations (Operator C)

The network operating landscape is dynamic and the operators must continually modify the networks to keep up with the situation and maintain a proper network fitness. Currently, it is done by manually adjusting many low level parameters and using low level tools, which requires a lot of manual effort. Parameter optimisation is considered difficult and due to the low experience level of the technical staff, many employees do not know how the features work. Hi-Tech Inc.'s system has more features and parameters than other suppliers networks making it more demanding for the operator. Particularly for an inexperienced operator, handling of the complicated system is difficult. The means to reduce the user-observable complexity should be studied. Particularly, the complex adaptive system approach would be worth exploring, as it has a great potential in eliminating the manual parameter optimisation.

There are very many separate applications packed in the NMS. Reducing the number of separate applications and designing them to better support the operator tasks, would improve the system usability. Also raising the perceived operating level, as seen by the users, would be required. This would constitute a metasystem transition in operating work. Exchanging data in both directions between the NMS and planning tools is necessary, but currently inconvenient. Easy data exchange would enable many useful, new functions and improve operating efficiency and thus system usability.

A local planning office of Hi-Tech Inc. has developed many tools that are used in numerous projects throughout the country. Many times, the real working solutions for planning tools are created in the local planning offices, where the projects are run. What is the reason why the global tools cannot compete with the 'quick-and-dirty' applications? There could be at least two major reasons for this. 1) The global tool development does not use close enough contacts with the real users and hence does not know the 'real-world-problems'. 2) The Stage-Gate development process cannot use even a modest amount of feedback from the users after the specification is frozen. After the long development without user feedback, the product is not likely to be what the users want. In other words, the weak coevolutionary links cannot provide enough selection information for the developers, and the products flow down on the fitness hills.

It also seems that the global tools are designed for 'green-field planning', new projects starting from scratch, but majority of the work is extension planning. Network planning efficiency could also be improved by creating a mobile access, both to the planning system and to NMS data. Nothing like this presently exists. These things also tell about weak coevolutionary links.

Hi-Tech Inc. has taught inefficient operating models to new operators by using direct access (MML) to BSS and not NMS as they should. Many times, NMS is not installed early in the project so that it could be used from the start. If people learn to use MML in the beginning, it is difficult to change to another way requiring a totally different mindset. As there are no recommended operating task procedures, the system cannot be optimally designed for any tasks. Suppliers could derive efficient operating procedures from well working examples learnt by research, so that they could be recommended to all operators. The system could also be adapted to fully support the recommended procedures, although this does not eliminate the need for flexibility in the system. Close coevolution is required.

The product development within different product lines has been quite independent. Traditionally, one product line has designed the base stations and implemented every conceivable alarm in the BTSs. The BTSs send all the raw alarms to the NMS without any correlation, which means that the NMS is flooded with alarms. To ease the situation somehow, NMS product line has designed alarm filters to reduce the number of alarms. This tells about lack of internal coevolution between the product lines. The mentioned direction of development has actually been changed already and the new BTSs do not send all possible low level alarms to the NMS, but do some diagnostics locally. This approach resembles the natural principle, where complexity is autonomously handled locally.

The operator has designed some tools of their own to improve the system usability of Hi-Tech Inc.'s system. For example, they have created a tool for data consistency checking after downloading the data into the network and adjacent cell information checking on the OMC border area. They also have a network management system of their own under development. As the dynamic landscape calls for fast tracking in order for the operator to stay on a fitness hill, they cannot rely only on the time-consuming development of the supplier.

6.11 Operator D

Operator D is a very experienced, advanced operator, who is using NMS a lot in their operations and they spend a lot of time for thinking how things could be done better. They are continuously looking for improvements for their current operating task procedures. However, their network is relatively small.

6.11.1 User-observable complexity

By adopting the task oriented thinking of the operators, it would be possible to integrate some of the tools and reduce the number of different applications. Currently, there are so many tools that it is a problem for many people. For example, there is only one person in Operator D, who knows all the NMS applications and the rest of the NMS users know maybe only a half of them. ***This is user-observable complexity.*** The user tells that when integrating a site, they use the Import-tool, Download and LAPD[a]-manager, RNW[b]-manager and NW-editor. *In other words, the operators are using several tools for each task.* Sometimes, the tools must be used separately too. Wizard-type application combiners would help a little, particularly the new employees. For the experienced employees, the impact would not be very substantial. *A metasystem transition and a restructurisation based on the tasks could be more useful.*

Each feature of the network in itself is reasonably understandable for an experienced person, but when you start combining different features in the same network, the things start to get difficult. *This is complexity by definition.*

Parameter handling (modifying) seems to be taking most of the effort nowadays. The NMS users have to execute about 100 change files/month and check that everything is working after the change. *The need for continual parameter modification shows that the landscape of everyday operating is dynamic. However, the amount of effort also depends on the target quality.* Of course, the network planners are involved too, because they have to design the new parameter sets. There are different default parameter sets for GSM900 and GSM1800. Also, adjacency parameters have different sets for 900 => 1800, 1800 => 900, 900 => 900 and 1800 => 1800, which are all stored in their own tool. Before this tool was in use, it was very difficult to handle the different parameters in NMS. The tool also makes a consistency check every night by comparing its own parameters with the parameters in the network and creates a list of inconsistencies. It is then possible to load the "correct" parameters from the tool to NMS and further on to the network. *Also this operator has circumvented a major problem in the system by developing their own tool. They cannot wait for the supplier to improve the system.*

6.11.2 GUI vs. MML

The GUI is slow in many applications. Skilled users can do many things faster with MML, for example, state changes. When doing them in GUI (NMS), you first have to open the right view, then drag and drop the data to RNW-manager, before you can actually do it. On the other hand, when creating and deleting adjacencies, it is easier with the GUI. When working with MML, the user is very close to the HW and uses various HW IDs, whereas with NMS the user does not even know these IDs. According to the interviewee, MML should normally be kept away from the users

---

[a] Link Access Procedure on the D-channel.

[b] Radio Network.

and handle the every-day-work with GUI. However, after each BSS release, when NMS support is not available for the new features, they are forced to do things with MML.

One of the down sides with the GUI has been that some users do not completely trust it, because there is no feedback and the status of the elements is not visible all the time. In the environment, where Operator D works (using their own additional tools), users trust the GUI without the good feedback. However, the fault codes are not very informative – they should be improved.

The level of working in every day operations is quite low. Designing some tools for higher level management of the network would be very useful. **The interviewee is suggesting a metasystem transition.** For example, how to manage the (dynamic) traffic load efficiently is a challenging task, as there are no tools for that. Currently they have to try to handle it manually with MML.

6.11.3 Synchronisation of subsystem releases

According to the interviewee, the lack of synchronisation in subsystem deliveries is clearly a problem. For example, when Operator D had to take the Complex_Feature in use, the feature existed in BSS but not in the NMS, so all the numerous parameters had to be handled with MML. When doing this, the possibility of making errors is huge, not to mention the required effort. Operator D decided not to use the Complex_Feature after the initial experiments. **This is an example, where the selection eliminated the product due to poor fitness. Coevolutionarily, the feedback means that the result is not acceptable and should be improved. With good system usability, the elimination could have been avoided. The system usability should be rather good from the very beginning, it is no use to start improving it after the customers have decided not to use the feature.**

Operator D creates their own tools to support the BSS features before the NMS support is available. The operators need to be dynamic and come up with solutions fast, so they have to be creative and invent their own ways of accomplishing things. **Effectively, the operator is saying that the landscape is changing faster than the product development of the supplier can follow. Consequently, they have to create some of the solutions by themselves.** Obviously, things can be done this way too, but it requires extra effort from the operator. Anyway, NMS is already going for different levels of releases. On the lowest level there is the basic HPUX[a]-layer, then on top of that

---

[a] Hewlett-Packard Unix.

there is the system support layer (e.g. GPRS[a] or some other standard technology) and the uppermost layer contains the features having the operator requirements. *This hierarchical structure resembles natural complex systems.* With this kind of structure, it is possible to create and manage the operator features without changing the rest of the layers, which makes the development faster and simpler. *This enables better tracking of the dynamic landscape for both the supplier and the operator.* It would probably be a good idea to use the same approach in BSS-development too.

6.11.4 Communication of future releases

When a new release is being considered, Operator D checks the supplier's documentation to see what new is available. On the BSS side, Hi-Tech Inc. is usually discussing the next two releases with the operators, but there is no sign of this on the NMS side, although it would be very useful. NMS product line does not provide any information about the next releases. For example, when Operator D was preparing a budget for –99, they asked information about the features of the upcoming release, but is was not given – not even a list of optional features. *This is a coevolutionary link breakdown. It also affects the operator's dynamic landscape tracking.* It would be very useful to see the new applications in practice before making the purchase decision. For example, if the new user interfaces were on the Internet to be seen and tested, it would be great. Good documentation and audiovisual multimedia presentations about the functionality of the network features, as well as NMS user interfaces on www would definitely be helping the situation.

6.11.5 Automation

When designing new ways of managing the network, it is difficult to decide what functionality to use[b]. Perhaps by automation it would be possible to avoid burdening the personnel excessively with the manual effort and complexity. For instance, if the network measures the call quality and traffic load and then creates new configurations based on that information, the manual work is dramatically reduced. It would also be good to get rid of permanent resource allocation for different purposes (like signalling channels, data, voice etc.). This could, once again, be done by measuring the situation in the network and then automatically adapting the resources to the dynamic situation.

---

[a] General Packet Radio Service. Mobile service, which gives a packet switched access over GSM to external data networks with high peak transfer capacity.

[b] Hi-Tech Inc. has the same problem with the numerous feature candidates. Only about 10 % of the candidates can be implemented. This is based on discussions in the work community of Hi-Tech Inc.

This has been studied in Neelakanta/Deecharoenkul 2000[157]. The outcome would be an optimum allocation of resources for each cell and each service. *This fits well the description of complex adaptive system.*

When starting to improve the system usability, it is good to start from tasks, which require most effort. As parameter handling is the most laborious area in operating, it should be taken under scrutiny first. The supplier's BSS product development has been developing the BSC – MS loop, but for automation purposes it would be good to have another, bigger loop consisting of NMS and some other tools (see Figure 38). The basic idea is that the operator would only define some high level target numbers, like call drop rate (e.g. <1%), utilisation of channels (e.g. >80%) and blocking (e.g. <1%) leaving the parameter optimisation to the automatic control loop. *The operator is suggesting a metasystem transition. All this is the very domain of complex adaptive systems.* This second closed control loop has a lot of potential to squeeze everything out of frequency efficiency. There is probably not so much potential left in the conventional BSC - MS loop.

Site integration used to take 3…4 hours, when it was done manually. Now it is done semi-automatically with the aid of their own tools, so that parameters can be transferred to NMS much easier and hence the integration takes only about 1 hour. NSS split tool would probably be very useful for bigger operators, but since Operator D is relatively small, they do not have to split the NSS very often.

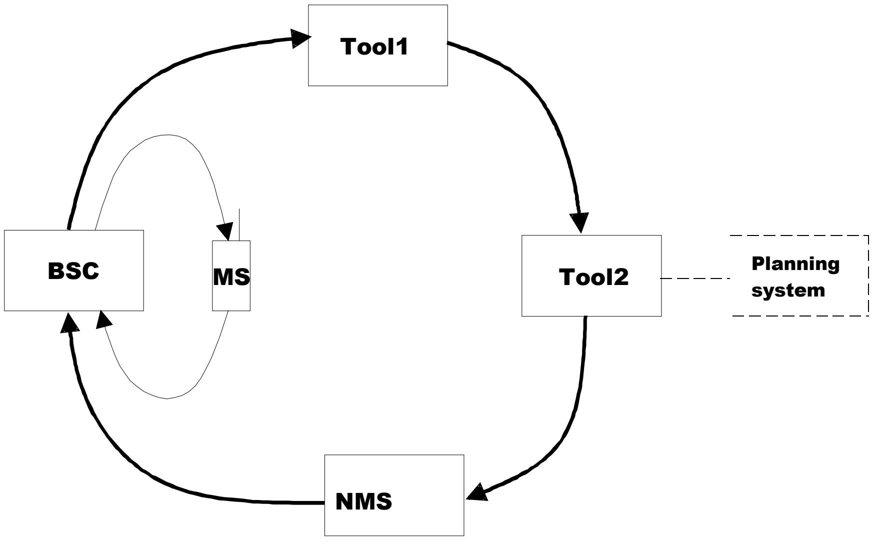


Figure 38. A second closed loop for automating the BSS. Traffic to the MS is relayed via the BTS, but since control wise the BTS is transparent, it has been left out of the picture.

6.11.6 System usability

A baseline for system usability could be established by first defining some major tasks that the operators are doing and then checking the man-hours used for these tasks. The situation should be checked again later on to find the reduction in man-hours and this way the improvement in system usability. The weighting factors of the formula (see Equation 2, in chapter 2.3) should, perhaps, be defined based on the load and frequency of the tasks, maybe also the severity of consequences. The latter one means that e.g. errors in network planning have more serious consequences than errors in building. System usability should be integrated into the operating procedures and tools so that the procedures carry through with good system usability. **This would require recommended best practices. Strong coevolutionary links between the operator and supplier are required to achieve this.** The significance of system usability will be emphasised in the increasingly dynamic operating environment of the future, because operators have to rapidly introduce new services to be competitive. In this kind of world, the operators having a network with good system usability have the upper hand. **According to the interviewee, the changing speed of the dynamic landscape is increasing.** The interviewee tells that operators would probably be willing to change their operating task procedures if better ones are recommended. Operator D is not against spreading information about efficient operating process, even if most of the information is coming from them. **The operator is willing to strengthen the coevolutionary links.**

Service life cycle management is reality already at least to some degree, but perhaps the life spans are so long that no services have been killed yet (-99). Some services have quite few customers though, and they are considered to be dropped. There has been some limitations to introducing the new services, but the limitations are mainly in the MSC and in the IN (Intelligent Network) system, not e.g. in the BSS and NMS. Sometimes, they have to make a workaround solution or use the second best idea to circumvent the limitations. It has been difficult to monitor the services, to see if they are up and running, since that kind of functionality is still missing. **Service management does not exist in practice.** Ideally, system usability should be so good that all-new services could be introduced, as well as other measures could be done, with little effort. This might be difficult to achieve, because it is hard to anticipate all the future services and things that should be possible. This places more emphasis on fast and flexible product development on the supplier's side. **Fast tracking of the dynamic landscape is required.**

6.11.7 Summary of interpretations (Operator D)

The users have to use a multitude of different tools, which do not follow the operating tasks very well. The operator is also looking for means to simplify their daily operations and reduce the user-observable complexity. Parameter handling takes a lot of effort, and particularly when many different radio network features are combined the things get very complex. As the operators have to change the network behaviour and they are forced to do that by manipulating all the numerous low-level parameters, it can be a laborious and frustrating experience. Default parameter lists are one way of handling the parameter manipulation, but there are also many different default lists for different situations. To ease the work, the operator has created their own tool for storing and handling the various parameter lists. All this indicates that the operating landscape is very dynamic. In this dynamic environment, where the numerous parameters must be frequently optimised, the operator would prefer just defining some higher level goals and let the network do the adaptation to the requirements. This metasystem transition could be realised by applying complex adaptive systems in the radio network.

NMS is a large collection of different tools, which are not quite efficient enough in the users' opinion. By complementing the system with their own tools, the operator has been able to run the operations quite well, although they recognise that the efficiency of the tasks could be improved in many ways. For example, the users would like to see higher degree of automation and use higher level management than the current system enables. They have also improved the system usability with their own applications by creating highly automated tools. This kind of metasystem transition approach would also be advisable for the supplier in the network management development.

Lack of synchronisation in the releases of the subsystems (BSS, OSS) is a problem and causes extra effort to the operator. The operator must create additional tools after each BSS release, because there is no NMS support available for the new features. This slows down the landscape tracking of the operator, as they cannot get ready made, system-wide products. It is even possible that the operators decide not to use certain features completely, if the handling is too difficult. The Complex_Feature experiment is a good example of that. The operator would also be interested in seeing a layered, modular structure in the radio network products, since it enables the development of the operator features independently of the other parts (reduces K-factor). In addition to being a good way to manage complexity, this also enables accelerated feature development thus improving the landscape tracking of the supplier.

The contents of future releases are not communicated well enough to the operator. It can even be difficult for the operator to make a budget for the next year, if they do not get information about the forthcoming products. Here, the coevolutionary links have been too weak. The operator would also like to test some products, e.g. NMS user interfaces, before release. The supplier should gladly accept this kind of free testing, which improves fitness of the product by providing selective information. This would be very effective coevolution.

System usability can be measured by man-hours, but the various tasks are not equal in the sense that the same effort spent on each of them would yield the same benefit. It would be good to integrate the system usability into the infrastructure by defining operating procedures first and then designing the system to fully support this way of working. The synergies between good processes and infrastructure adapted to the processes would amplify the usefulness of the technology (refer to Figure 18 in chapter 4.2.2 'Complexity catastrophe'). This is coevolution of operating processes and the system.

Service management is already being done in a way, but presently it is difficult to supervise the services, as the system does not support the monitoring of this end-to-end functionality. Anticipating all the different services of the future several years ahead is not possible. This means that developing systems that support the dynamic service life cycle management of the operators will be even more challenging for the product development of the suppliers than it is today. This will require strong coevolutionary links between operators and suppliers, as well as more flexible product development processes on the supplier side to enable fast tracking of the dynamic landscape. Of course, the system itself should also be flexible to allow changes in operating tasks.

7. SYSTEM USABILITY AS REVEALED IN THE EMPIRICAL STUDY

In this chapter, we shall review how the system usability of the present cellular networks looks like based on the field study. In the first subchapter, we describe the main findings categorised in five different categories rising from the theoretical framework: Interdependency, user-observable complexity, coevolution, dynamic landscape and the consequences of the underlying problems. In the three subchapters following this, we address some higher level implications of the findings.

7.1 Challenges of system usability development

The major findings presented in the following text constitute challenges, which should be met in order to improve the system usability of the cellular networks. In chapter 8, we try to find some ways to meet these challenges. The challenges listed in the tables are presented from the operator point of view, except for those in parentheses – they are only indirectly visible to operators.

7.1.1 Interdependency

Let us first study the findings that indicate the complexity of the system and the operating work (see Table 8). The system contains a large set of manually optimised radio network parameters, some of which are interdependent (refer to chapter 6.4 'Simulations'). This interdependence (complexity) complicates the parameter optimisation procedure. There are interdependencies also between various working groups and their tasks, for example, in the commissioning process, where the NMS-users and field personnel have to synchronise their work in order to complete the task. This makes the work complex and time consuming.

One could also say that the tools stipulate the way of working, since they only enable a certain way of accomplishing the tasks. In other words, there is interdependency between the tools and operating work. This could also be understood so that the supplier wants the tools to guide the users through the tasks. The idea would be sound in principle, but it requires a very good understanding of network operating. Many times though, the users considered those ways of working so awkward that they rather designed their own tools to accomplish the tasks. A similar discovery can be found in Lucas/Meech/Purcell 1997[166], where OMC functionality of a GSM system was studied. They found that advanced users had developed tools to enhance the functionality of the system.

There is also quite a large amount of work in software development on the supplier side, which is not directly visible to the operators. Developing the software packages takes a lot of time and effort. One thing contributing to the large effort is probably the non-modular software in the BSS. As the interdependency (K-factor) within the software is high, all changes generate many alterations elsewhere in the code.

Table 8. Findings related to interdependency

Interdependency of parameters
Interdependency of working groups and tasks
Tools stipulate way of working
(Laborious development with non-modular SW)

7.1.2 User-observable complexity

We shall now take a look at the findings related to user-observable complexity (see Table 9). There is a large number of different tools in use – some of them are integrated in the NMS, some are in-house developed and some are 3rd party tools. Since this multitude of applications has never been designed to work together, the working efficiency is not very high. Each tool also covers only portions of the tasks and the level of automation is rather modest. The supplier has not even tried to make the tools task oriented referring to the different operating processes of operators. This has lead to a situation, where the system does not properly support any tasks. Since the level of automation is low, performing the operating tasks requires a lot of manual effort.

Table 9. Findings related to user-observable complexity

Fragmented field of very many tools
Low level of automation
Complicated algorithms (user-observable complexity)
Many manually optimised, low level parameters
Many technologies, frequency bands, layers, services, vendors
Old features are not removed

The radio network algorithms are quite complicated requiring a considerable effort from network planners to comprehend how the features work and how they can be optimised. The user-observable complexity increases also because the (interdependent) functionality must be optimised by manually adjusting a large number of low level parameters. There are also different technologies (circuit switched, packet switched, single and multi-slot), frequency bands (900 MHz, 1800 MHz), cell layers (macro cells, micro cells, indoor cells) and services. As nothing is ever removed, the situation gets more complex after every release. There are warnings in literature against this kind of repeated layering (Brown/Eisenhardt 1998[184]).

7.1.3 Coevolution

In the following, we shall examine the findings related to coevolution (see Table 10). Usually, the suppliers do not operate cellular networks, and particularly the product development people of the supplier have a rather modest understanding of network operating. Although there are some people that have this understanding, these people are not designing the systems. There are no recommended best practices for operating task procedures either. This tells about modest coevolution with operators.

Presently, when the suppliers ask for feature suggestions from the operators, there are difficulties in deciding, which of the very many suggestions they would develop. Since the initial feature research does not dig very deeply in the operators' problem field, the selected solutions do not necessarily solve the underlying problems effectively. In addition, as the product development process does not allow any iteration during the development cycle, the released products are not quite what he operators expect.

The releases also come out at different times of the year and cause unpredictability in the eyes of the customers. Sometimes it also happens that the supplier does not reveal the contents of the forthcoming release, so that the operators could prepare their actions. Additional effort for the operators is caused by the fact that the BSS-releases and OSS-releases (NMS) are not delivered together, but the management applications for the BSS-features come later. This means that the operators have to create their own tools to cope with the new radio network features.

Table 10. Findings related to coevolution

Operating understanding not sufficient in supplier's R&D
(Difficult to decide which solutions to develop)
Light feature research
Unpredictability of product contents and timing
Separate subsystem releases cause extra effort

7.1.4 Dynamic landscape

Next, we study the findings related to dynamic landscape (see Table 11). The networks require constant care from the operators. For example, to keep the quality sufficient, the numerous parameters must be tuned every now and then, particularly after major extensions. This requires a lot of effort in large networks.

In the light of the problems the operators are facing, it seems that the present product development process is not very suitable for this particular kind of business (complex hi-tech products), since it is not capable of producing the desired functionality with a relatively short notice. The present method does not allow any iteration with customer feedback within the development cycle. In addition, the rigid development time is rather long, particularly on the radio network side. This means that the required functionality should be known 2…3 years in advance, which is not possible. Since the landscape of the operators is dynamic, they cannot wait for the solutions as long as the development takes. Many times, they design their own tools for the tasks. However, they cannot create their own radio network (BSS) features, and have to leave that to the network suppliers.

As the radio network features are never redesigned, the original, non-optimal functionality is maintained in the system indefinitely. Presently, the network element management is the only management layer in the management systems and no higher levels of management exist. This means that the network management is done at rather low level and e.g. service management capability is still missing. For example, operators cannot see how the end-to-end services are running, since the systems tend to only show whether certain network elements are up and running.

Table 11. Findings related to dynamic landscape

Continuous optimisation need if good quality required
Dynamics of development process not matching the business
Lack of iteration in product development process
Operators cannot wait for the solutions
Radio network features are not redesigned
Service management capability does not exist

7.1.5 Consequences

Finally, we shall study the findings related to some consequences of these phenomena. As mentioned before, various operators have a little different ways of working. Because the supplier has not defined any recommended operating task procedures (best practices), the system has not been adapted to any operating tasks. This means that the operating efficiency leaves room for improvement. Since the manual optimisation of a large parameter set in a changing environment is so labour intensive, the networks are never in a true optimum state. This also means that the networks are not running with full capacity.

Many times, the operators do not get the wanted functionality at all, because the supplier can only produce a small fraction of the features requested. Even in the case, where the operators do get the suggested features, the waiting time is quite long. Considering the dynamic landscape of operating, the long delay makes the landscape tracking very difficult. Due to the light feature research and rigid development without feedback, the solutions are not optimal when released. Since the radio network features are never redesigned after the initial release, the situation is note getting any better even in the long run.

Table 12. Findings related to the consequences

System not ideal for any operator processes
Networks never in true optimum
Only a small fraction of feature candidates realised
New features come too late
Solutions not optimal at release
Many non-optimal features in the system

Product development of complex technical systems is a demanding mission. Although, we cannot expect to find the global optimum in a complex system, we should be able to find excellent solutions. To find these excellent solutions, we need complex system understanding. All the described challenges cannot be met by applying the present usability paradigm and improving the system user interfaces, but instead it requires a more comprehensive approach. If the network complexity is left untouched by usability considerations and the user-observable complexity is increased uncontrollably, the damage done cannot be mended by improving the user interface issues alone. Some of the effort presently used for the mentioned usability issues could be better spent by applying system usability thinking throughout the system.

7.2 Complexity in cellular radio networks

In these next three sub chapters we take the analysis on a little higher level to see what the results actually mean. It seems that the complexity issues have not been recognised and integrated in product development guidelines. Both the effective and random complexity in cellular networks have risen quite high, and since there has not been enough metasystem transitions, the user-observable complexity makes things difficult for the users.

In fault management, the principle has been to send all the low level fault alarms from all network elements to the network management system, which means that the NMS is flooded with tens of thousands of alarm messages every day. As we can recall from the theoretical framework, a better way would be to use rather autonomous layers and govern the complexity locally in each layer. The latest development has actually been towards this, as the newest base stations perform some diagnostics autonomously without sending all low level messages directly to a central location.

The lack of complexity considerations has also lead to many interdependencies (high K-factor), for example, in the radio network parameters. This makes parameter optimisation even more difficult for the users, as the parameter changes do not cause the same effect every time due to their interdependence. The interdependency should be considered when designing the algorithms, so that the K-factor is not increased excessively. The interdependency is also high in the BSC-software architecture due to lack of modularity. This means that every change in the code generates many alterations elsewhere, thus increasing the work loads of designers. As nature has demonstrated, usable complex systems are developed by using a modular and layered structure.

As we can recall from the theoretical framework, the metasystem transitions are a way to administer the ever-increasing user-observable complexity by isolating the users from the lower layer complexity, every time the system gets too difficult to handle. As the users access the system via the new metalevel, they are not overwhelmed by the technical details. The metalevel transitions can also be smaller, not involving a total restructuring. It is also possible to gradually reduce a) the random complexity and sometimes even b) effective complexity within a metasystem layer, when designing the systems and operating procedures. For example, redesigning a handover algorithm so that many different handover types are combined into one, simultaneously eradicating half of the parameters, would reduce the effective complexity (and thus the user-observable complexity), although still maintaining the functionality. Particularly, creating new superior solutions with lesser effective complexity requires good understanding of the operating problems, which can only be attained by close coevolution with operators.

7.3 The different worlds of operator and network supplier

The challenges presented above point out differences in the standpoints of the suppliers and the operators of cellular networks. Operators and suppliers approach cellular networks from different directions, which seems not to be completely understood by the suppliers. Consequently, there is not enough operating understanding in the supplier's product development. In the long run, the situation could be improved by enhancing the coevolution between the suppliers and network operators. One of the first steps in improving the operating understanding could be that suppliers adopt a more ethnographic methodology for learning about the operator world. As Wilcox 2001b[186] puts it, a key goal of ethnographic methodology is to understand a foreign culture, customers and users, in effect constitute foreign cultures vis a vis product developers. Operator world is indeed different from the world of technology suppliers, and suppliers require a good methodology (e.g. ethnography) for understanding this alien culture. In this sub chapter, we shall study the different standpoints to network management and radio network features.

7.3.1 OSS applications

Due to the different frames of reference, the network operators and suppliers have somewhat a different view to utilising the network management tools. In the technology producing frame of reference, the engineers like to create all kinds of devices. The problem, though, is that the very endeavour of developing products creates the potential for an egocentric subculture that is very different from that of the customer (Wilcox 2001a[167]). The product development of the technology suppliers has an *application point of view*, whereas the operators have a *task oriented point of view*. In other words, the suppliers produce SW-applications (tools) based on their own technological frame of reference and then more or less hope that the operators could use those applications in their daily operations, whereas the operators think in terms of what must be done to accomplish their daily work tasks. ETSI specification EG 201 472 2000[168] defines *tasks* the following way: Tasks are the activities undertaken to achieve a goal.

There are three main things underlying the situation. a) Modest operating understanding in design departments, b) Fragmented view to operating tasks, c) Different ways of operating among operators. The supplier might have some people with operating understanding in the company, but they are not in the design departments – they are working in projects helping new operators to cope with the operations. The designers have a very limited view to the tasks, as most of them never see the operating tasks in practice. The designers also work on small parts of the management system and hence do not see the big picture. In addition, there is no common way of operating a cellular network – operators have a little different ways of doing things.

The consequences of all this are visible in the daily operations of the network operators. The numerous tools are not very task oriented and also always lack some functionality. Low degree of automation forces operators to work at quite a low level by using many manual operations and many details. Effective tools, particularly for mass operations, are missing. Operators also produce many tools of their own and buy some third party applications. This has lead to a situation, where the operators have a huge number of different applications to manage their daily work (see Figure 39). The efficiency is not very high, as this complex of different tools has never been designed to work together.

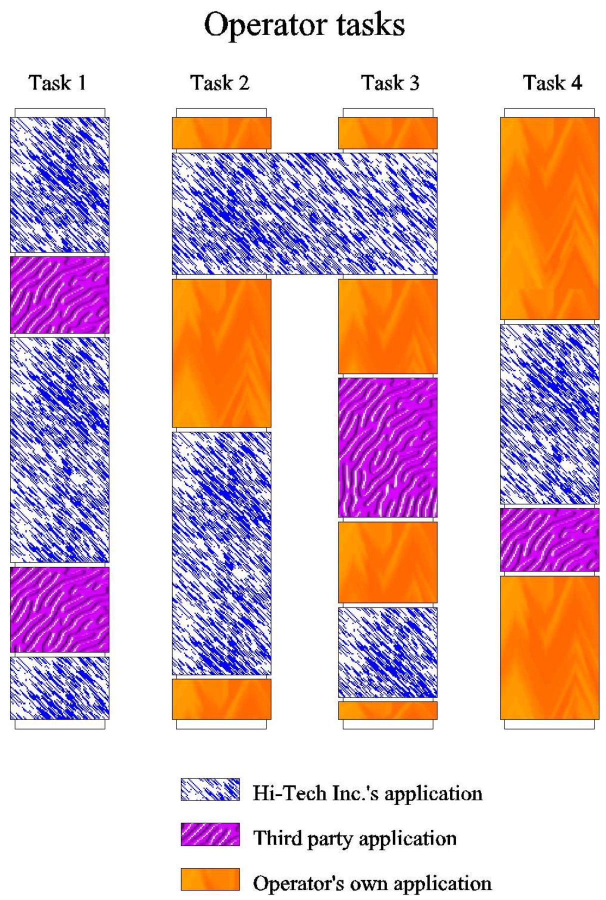


Figure 39. The fragmented world of operating applications (tools). The operator tasks are on the vertical columns, which are covered by the rectangles representing the various applications used for the tasks.

It also requires some effort from the operators to design efficient ways of working, when the mixture of tools and applications is not very flexible to allow straightforward task oriented operation. Consequently, the processes of even some of the most experienced operators are not very efficient and leave room for improvement. On the other hand, the operators never get what they want, since the suppliers have to make a compromise between the contradictory requirements. All this shows that the coevolutionary links between operators and suppliers have not been strong enough to bring the cultures close together. It would be a mutual benefit to improve the coevolution, as fitness of all players improves in well working coevolution.

7.3.2 BSS features

The same kind of duality in thinking as in the OSS applications, also seems to exist in the BSS features. This is probably caused by the cultural difference and the hasty product development schedules of the supplier. The supplier has to implement the functionality down to the little details and hence the attention of the designers is focused on the low level. This easily leads to low level controlling means, and if the schedules are tight, no effort is invested in additional design to create higher level control. The operators, on the other hand, approach the situation from the other direction, as they have their operating goals to fulfil. They think in terms of operational tasks and often find that accomplishing the tasks with such low level means takes too much effort.

The suppliers produce networks with high user-observable complexity. In addition, there is interdependency between many network features. Even the suppliers do not have a sufficient understanding of the combined effects, which means that the operators must explore the phenomena in practice. The interdependence of many operating tasks add the operating complexity.

There are two basic reasons for the large variety of manually optimised control parameters in the system, the number of which has been increased in every new release. 1) The system designers have always thought that the control possibilities in BSS must be very versatile, and hence they have created all these different manually tuneable parameters. 2) The system designers do not completely understand the complicated phenomena underlying the new technical features, and since there is a pressure for rapid feature development, they have left those problems to the users. Consequently, they have created a set of parameters, by which the features can be tuned to work in the field with a considerable manual effort. These reasons were also independently verified by an experienced BSS designer and another experienced employee of Hi-Tech Inc. In addition, one product development manager in the company confirmed that suppliers typically cannot say how different things affect each other. Operators create their own rules of thumb as they go along.

From operators' point of view, the less manual work is required in the field the better. Generally, they are not interested in adjusting the low level parameters, as long as the network functions decently. They avoid optimising the BSS parameters and use default parameter sets whenever possible. Still, operators must be able to control network behaviour somehow. Higher level control via metasystem transitions would enable significantly reduced user-observable complexity and manual effort. Operators would like to shift the effort from maintaining and optimising the network to managing the actual services.

Enhancing the coevolution between operators and suppliers would narrow down the gap between the worlds. Some means for achieving this are explored in chapter 8 'Using the theoretical framework to find guidelines'.

7.4 Coevolution in product development

The product development uses information from three basic sources: 1) Standards, 2) Research and 3) Customer feedback. Representatives from both the suppliers and the operators take part in creating the telecommunication standards and thus affect the future development on high level. As this is the earliest possibility to influence system complexity, it might be a good idea to include some complexity guidelines in the standards. After all, the biggest mistakes are usually done in the early phases of development. In any case, the standards leave a lot of freedom to the technology suppliers. The suppliers have their own ideas and technology research, which are used in developing new technologies for the market. Quite often this 'technology push' is necessary, since the operators do not know what the best technical solutions would be – particularly, long before they are required. The most interesting coevolution as far as this research is concerned, however, occurs after the standards have been created and the products are developed with the aid of customer feedback. In this chapter, we shall first list some arguments for the technology push and then concentrate on coevolutionary product development with the aid of the customer feedback.

Many times, the improvements coming from the operators are not revolutionary. There are certain reasons behind the conservative improvements suggested by the operators. For example, operating personnel's ability to guide the product development is limited by their experience and their ability to imagine and describe possible innovations. If no current product exists having the new features even in the most primitive form, the users have no foundation on which to formulate their opinions.

They are also so used to the current conditions that many times, they do not ask for a new solution, and even when they do, the improvements are conservative and lack the comprehensive view due to their limited task field (Leonard/Rayport 1997[169]). Many times they do not even know what they want (Lester/Piore/Malek 1998[170]). For the mentioned reasons the technology push by the suppliers should not necessarily be understood as a negative thing. The suppliers must innovate new technical solutions, but they should also have a good understanding of the operating domain in order to do so.

The method for gathering the information from the customers also plays an important role. When users' needs are solicited in concise written form or through constrained dialogue and delivered to product developers in compressed form, critical information may be missing. Consequently, the standard techniques of inquiry in new feature research rarely lead to truly novel product concepts (Leonard/Rayport 1997[169]). Therefore, a great care should be taken when doing research for new features. A suggestion for a method of feature research is described later in the chapter 8.3.1 'Empathic feature research'.

The actual product development cycle takes 2…3 years to complete. As described before, the Stage-Gate process (refer to Figure 28), which was originally designed for a rather slow-paced industry, is not able to use any customer feedback allowing iteration within the development cycle. The outcome in a dynamic business is:

- Those features that were implemented, come after a considerable delay
- The outcome is not quite what was wanted, since no iteration is used
- All modifications must go through the development tube again

The results indicate that the operators' landscape is so dynamic that they do not want to wait for the solutions for 2…3 years. Many times, the operators create their own tools in the mean time. When the product comes out of the development tube, the result may not be quite what the customers want. This happens, because the supplier did not completely understand the user requirements in the first place and because the customer feedback was not used during the long development. After the release, the customer feedback is collected, and some of it is used for modifying the product. Unfortunately, the customers can get the modified product after another 2-year delay, during which the situation may have changed so much that the product can be obsolete. In the present situation, iteration during the development cycle is not possible, and hence the opportunities for utilising the selective force to drive the solutions to the fitness hills are wasted. Consequently, the rigid product development process prevents fast tracking of the dynamic landscape. Improving the customer input in product development is addressed in chapter 8.2.1 'Product development process'.

8. USING THE THEORETICAL FRAMEWORK TO FIND GUIDELINES

In this chapter, we utilise the things learned in the theoretical framework to generate solutions for the challenges of system usability found in the empirical study. However, these solutions are not meant to be directly implemented, they are guidelines for the engineering organisation, so that they can create the implementations. The empirical study of this research has been done from the network operator's point of view. However, as the system usability is created by the technology producer, the results are required on the technology supplying side. Otherwise they cannot be utilised for improving the system usability. Therefore, we formulate the solutions in the way that they can be beneficially used by the technology developer. These guidelines have been listed in three different categories: Complexity related, dynamic landscape related and coevolution related. There is also a concise list of the guidelines in chapter 9 'Summary'.

8.1 Complexity related guidelines

8.1.1 Reduction of user-observable complexity

Increased effective complexity gives an entity a higher potential to be more sophisticated than less complex ones. Complexity of construction makes simple things more complex in order to make them fulfil additional constraints (high performance, usability under extreme conditions, safety, etc.) (Schuster 1996[101]). A more complex biological entity, e.g. a human being, is potentially able to achieve a wider variety of goals than a less complex one, say, a worm. However, this potential must be usefully realised to avoid unnecessarily complex manifestations, since there is always a cost associated with complexity increase. This cost is visible both in constructing and using the system.

As we discussed in the theoretical framework (chapter 3.1 'What is complexity?'), the user-observable complexity can be divided into two components: Non-random (effective) and random complexity. We also noted that it is possible to decrease both of these components even without reducing the functionality. Any process that decreases the system's total information (effective + random complexity) increases one's ability to describe and control the system (Gell-Mann/Lloyd 1996[117]). Hence, it would be useful if we could minimise user-observable complexity.

The effective complexity is considered to be a positive thing, although even that can be reduced by radical innovations, which still maintain or even improve the performance of the system. Unfortunately, the innovations that are meant for reducing the complexity are many times discarded as useless in industry, since they do not seem to provide any new functionality. The random complexity, on the other hand, is just straining the user and should be minimised by design (and gradual redesign) to avoid excessive burden to the users. It is also possible to 'isolate' the user-observable complexity from the users by using metasystem transitions (see chapter 8.1.2).

The relationship of user-observable complexity and system usability can be demonstrated with an example. This example has been adapted from Kauffman's biological example (Kauffman 1992[126]). We could think that when the number of components (N) in the system increases, also the maximum possible system usability (also N) increases without bound. However, in the maximally coupled case, K=N-1 (K=cross-coupling of the parts, see chapter 4.1 'The NK model in engineering'), the total system usability increases ever more slowly due to the conflicting design constraints (see chapter 4.2.2 'Complexity catastrophe'). If we suppose that the cost per part is constant, the total cost rises linearly. In other words, every additional part adds the cost the same amount but the system usability less and less. Consequently, at some point total cost exceeds total system usability and further increase in complexity is no longer profitable. How soon this point is reached, depends on how much we invest in complexity management, for example, to reduce random complexity by design. With little or no effort in complexity management, the limit is reached sooner, and after that we increase complexity and cost much faster than benefit for users. However, as mentioned elsewhere in this document, this can be avoided by reducing the interdependencies (K-factor) of the technology e.g. by modular design and hierarchical structure. It would also be advisable to a) remove random complexity, b) reduce effective complexity occasionally and c) use consecutive metasystem transitions, to keep the user-observable complexity from rising too high.

8.1.2 Metasystem transitions

Today, the degree of automation is not very high in the cellular networks. There are some individual applications in the Network Management Systems that have a modest automation level – in the BSS the automation is even lower. Many operational tasks require long manual procedures with many different tools and a lot of co-ordination with other work groups (see the example process in chapter 6.8.4 'Commissioning', Figure 34). The radio network control algorithms have very many parameters all designed for manual optimisation, and the number of the parameters has been increasing year after year. This system, which is controlled at quite a low level by the different users, would be easier to handle if the control level could be raised.

In this framework, automation is related to the concept of *metasystem transitions* (Turchin 1977[100], Heylighen 1994[98], Heylighen 1999[99]) (see chapter 3.3.1 'Reduction of user-observable complexity'). The basic idea in metasystem transitions is that every time, when the user-observable complexity of the system exceeds a certain level, we create a new metalevel and hide (e.g. by automation or other architectural method) the technology below that. If the natural principle is followed, the subunits are already very autonomous and there is no need to do extensive re-engineering on the lower levels. By following this principle, the user-observable complexity (at the metalevel) can be kept relatively low, even though the system itself is getting more complex all the time. Tang/Salminen 2001[115] uses the concept of complicatedness, which is very similar to user-observable complexity. According to them, the system is designed to have a certain optimal complexity, which matches the required functionality. This optimal complexity conforms to certain complicatedness. Metasystem transitions are meant to reduce this complicatedness without affecting the effective complexity.

8.1.3 Complex adaptive systems

In the present paradigm, where the whole system – including the software – is rigidly constructed before taking the system into use, the variations in the operating conditions are handled by manually adjusting a large number of parameters. Many times, this tuning requires frequent and quite a substantial effort. In addition, the pre-stored possibilities in the system do not give a complete control. As the Ashby's law of requisite variety states, in order to achieve complete control, the variety of compensatory actions a control system is capable to execute must be at least as great as the variety of perturbations that might occur (Ashby 1958[171]). In the world of infinite variety, this is not possible without adaptation. In such a situation, a good alternative could be the technique called *complex adaptive systems* (Gell-Mann 1994[114], Holland 1995[124]). Using this approach would mean quite a radical change to the conventional engineering, and hence require a totally new mindset. Creating this technology is not a trivial task and requires a lot of research and development, but the potential rewards are also high.

In this technology, the designers do not create the actual solutions, but instead they create a system capable of finding new solutions all the time. In a way, the system searches for fitness hills, climbs to the top and stays there. As the adaptive landscape is dynamic, the system tracks the variations, thus automatically adapting to changes in the environment. Adaptability means that you do not have to store all adaptations in advance, the system adapts as it finds out how the environment changes. This avoids the need for the system to possess all of the requisite information in advance and it relies on environmental cues to fill in the details (Christensen/Collier/Hooker 2000[172]). In other words, reality is spontaneously unfolding at the present moment, instead of manual selection of options from the pre-stored possibilities.

As discussed earlier, adaptability is characterised as the capacity to engage in interaction with the environment, thus generating new information in the system (ibid.). This is done by creating and continually modifying rules that describe the regularities governing the studied phenomenon (Gell-Mann 1994[114]). In a cellular network, the system must measure certain things of the radio path (e.g. signal strength, quality, interference etc.) to form a picture of the real world. Then the various elements of the system adapt by combining a tentative schema based on the history with the incoming information, thus creating new information for changing its behaviour. Correctly predicting schema is retained and assigned a high value, whereas contradicting schema is discarded (ibid.).

8.1.4 Modular software structure and hierarchical architecture

One major thing contributing to excessive complexity is the lack of modularity in the SW. In a monolithic SW-package, everything is connected to a great many of other entities thus increasing the K-factor, which denotes the degree of internal cross-coupling in the software (refer to chapter 4.1 'The NK model in engineering'). If the K-factor gets very high, this increasing complexity leads to the complexity catastrophe (Fleming/Sorenson 1999[129], see also chapter 4.2.2 'Complexity catastrophe). In other words, if everything depends on everything else, there are no optimal solutions, but all solutions lead to performance compromises. Modular design makes components less interdependent, thus reducing the likelihood of a complexity catastrophe. On the other hand, keeping up and developing a very large monolithic SW-package drains vast amounts of developing effort, as every change induces a dozen other changes (Ulrich 1995[173]).

There are many books and articles testifying for the benefits of modularity. For example, Baldwin/Clark 2000[174] argue that the industry has experienced the high levels of innovation and growth by using the concept of modularity, building complex products from smaller subsystems, which can be designed independently, yet function together as a whole. Baldwin/Clark 1997[175] tells about three types of visible design rules: 1) An architecture to assemble modules into a working system, 2) interfaces to connect individual modules, and 3) standards for testing conformity of a module to the design rules and for measuring the performance of one module relative to another. Schilling 2000[176] identifies and discusses factors affecting increases in inter-firm product modularity. Sanchez 1996[177] gives a general overview of the modular design approach and emphasises the importance of fully specifying interfaces between components (e.g., signal exchange between components) at the outset and keeping them unchanged during the development phase.

In addition to being modular, natural systems are hierarchical, handling the complexity of any part at the level in question. Thus, only those functions that require highest level control are centralised. For example, the cellular metabolism is autonomously handled at the cell level without centralised control, and immune system autonomously destroys harmful invaders (Christensen/Hooker 1999[178]). With this approach, we can avoid the very high number of links to the control point and also the need for extremely high processing power required for controlling all the functions of a complex system (refer to chapter 3.3.1 'Reduction of user-observable complexity'). This also allows smaller amount of interdependencies (K-factor), as many functions can be handled locally by autonomous subunits, and thus it reduces the danger of complexity catastrophe.

8.2 Dynamic landscape related guidelines

8.2.1 Product development process

With increased competition and constantly changing technological landscape, there is increased burden on the project teams to deliver a product that meets the customer needs in the first try (Upadhyayula 2001[179]). In order to find the excellent solutions in the vast solution space of complex technical systems, one must explore very many different product variations. In order to speed up the procedure and avoid expensive changes later, it is necessary to use the selective force from the markets. This means using the customers' insight early in the process – before the products even exist. The very first thing in the development is the feature research with customers in order to learn the problems and get ideas (see chapter 8.3.1 'Empathic feature research'). The next thing to do is testing the product variations. This is presently possible e.g. by using streaming multimedia and the Internet.

This screening process could be done by creating audiovisual multimedia descriptions for the forthcoming new features (see Figure 40) to provoke feedback from the customers. This kind of testing provides a large amount of very useful information very fast, and in this way provides the required selective force already *before* the implementation (Dahan/Hauser 2000[180]). The feature candidates described with animated audio-visual multimedia are placed on a web-server for testing by the lead customers. The first tests contain several different variations of the features, which then converge into a proposition for the actual features (a *virtual coevolution* before implementation, so to speak). After a few iteration cycles, the actual product development is started.

The time-constrained radical new product development consists of uncertain, ill-defined and unstable design tasks. This necessitates a flexible product development process, where designers can continue to change and to shape products even after their implementation has been initiated (Dragut /Bertrand 2001[181]). The iterative approach to product development is favoured because companies usually build better products, if they have the flexibility to change specifications and designs, get and incorporate market feedback in the products and continually test components as the products are evolving (Upadhyayula 2001[179]).

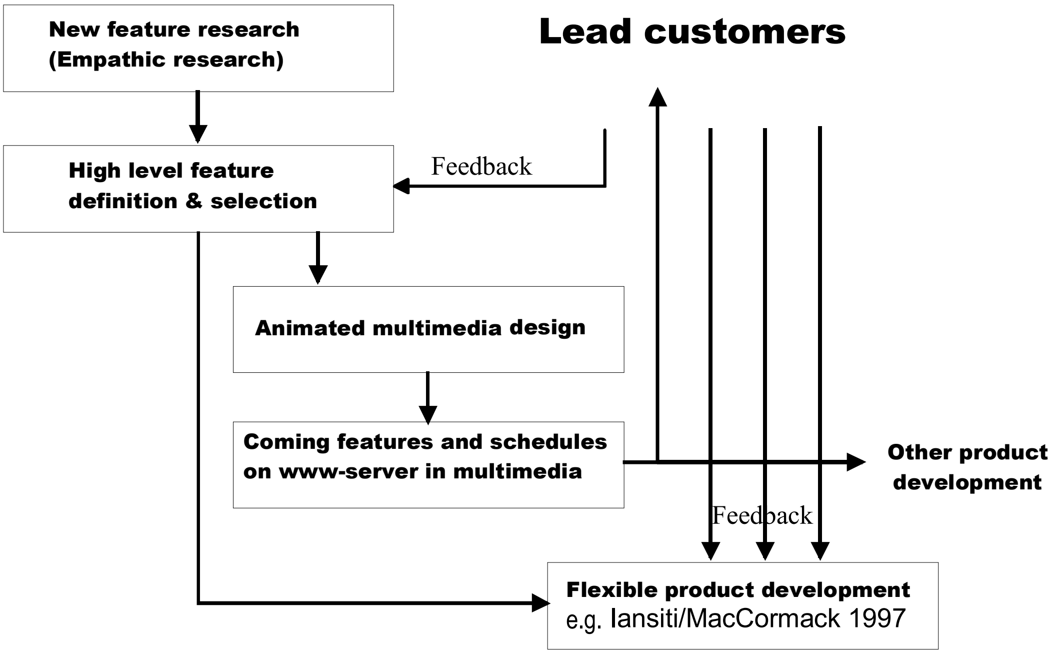


Figure 40. Improved product development process in the big picture. Animated multimedia descriptions enable 'virtual' coevolution before actual product development, as well as feedback for the flexible development process. Feature research is using empathic research method.

In the long run, it is also possible that the customers design their own products (Dahan/Hauser 2000[180]) by using the design facilities of the supplier via the Internet. In that case, the customers can choose the wanted features e.g. by drag-and-drop. Thus the supplier gets a ready-made specification for the product, which is then constructed from the modules. However, the supplier must still design and integrate the modules.

In the *user interface* design, it is possible to go even a step further by placing the newly developed GUI-*prototypes* on a web-server, so that the customers can try them already in the development phase and thus provide the means of iteration for the product development within the development cycle. This would be quite rapid coevolution compared to the traditional 2-year development cycle.

A flexible product development process is a prerequisite for system usability in the fast-pace telecom/internet business. If the product development is not able to keep up with the business (=track the changes in the dynamic adaptive landscape), even in the hypothetical situation where a product would start from top of a fitness hill (where the present process most likely cannot take it), fitness of the product would immediately start deteriorating (see chapters 4.5 'Dynamic adaptive landscapes' and 4.2.1 'Error catastrophe'). The dynamics of the product development process must match the dynamics of the business in question.

One flexible product development model was presented by Iansiti/MacCormack 1997[182] (see Figure 41). In the model, the concept development phase and the implementation phase overlap instead of following each other sequentially. By accepting the need of changes, companies are able to respond to a new information that arises during the development cycle. Systemic changes in the definition of a project and basic direction are managed proactively – designers begin the process with no precise idea of how it will end (ibid.).

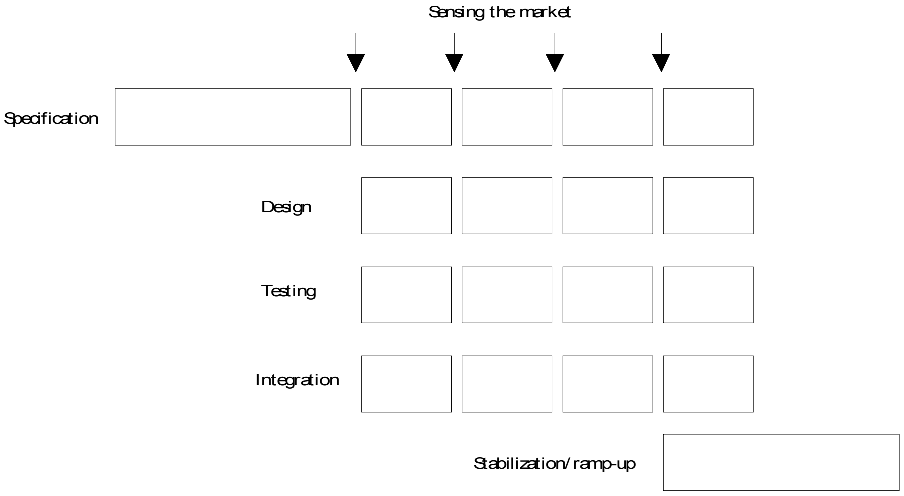


Figure 41. The structure of an iterative product development process (Iansiti/MacCormack 1997[182]). The specification and implementation phases are overlapping. Product changes are made with the aid of customer feedback.

The product development process is also affected by product modularity. This means that both should be considered simultaneously when re-engineering either the product architecture or development process. Processes can be structured around the product, modules assembled in parallel, testing can be done on individual modules, variety introduced late and thus orders rapidly fulfilled (Marshall/Leaney/Botterell 1999[104]). It has been shown that the main issues in product development are how to meet increasingly specific customer demands without the added burdens this can place upon development and production costs, time and quality. Modularity provides product variety to the customer. However, the variety can be offered efficiently through a limited number of modules and the use of common modules. Variety can also be introduced without unnecessary re-engineering, in reduced timescales and at lower cost (ibid.).

If we further study the SW-development in a network supplier company, we can see that the designers are fully reserved for these basic releases and if some major new ideas are introduced, it is impossible to implement them due to lack of resources. Probably, the best way to handle the situation would be to have two parallel development paths, revolutionary and evolutionary. As Gary Hamel has expressed this (Hamel 2001[183]): "Radical innovation and incrementalism must go hand in hand." The former could be handled by using external subcontractors and the latter by in-house designers. The evolutionary development should also include some degree of periodical redesign (Brown/Eisenhardt 1998[184]) to improve the features after the initial implementation (see chapter 8.2.2 'Gradual redesign'). However, the system usability of the features should be fairly good already in the beginning to avoid situations, where the customers abandon the feature after the first trial due to poor system usability (refer to the incident in chapter 6.11.3, where the Complex_Feature was rejected by Operator D).

8.2.2 Gradual redesign

Another thing of concern in the product development is the way the example company (Hi-Tech Inc.) has been adding new features on top of the old SW without redesign or questioning the system usability of the old part. This kind of "repeated layering" increases the user-observable complexity of the products quite rapidly, and there are warnings against it in the business literature (e.g. Brown/Eisenhardt 1998[184]). Refer to chapter 3.3 for definition of user-observable complexity (random + non-random complexity).

Eventually, the repeated layering becomes a necessity, because the SW becomes so convoluted that no one really understands it anymore. As a consequence, e.g. the product development times get longer and longer. This could be avoided by redesigning a particular portion of the product on each release, thus preventing it from becoming incomprehensible and obsolete. Hi-tech companies should have a re-architecture strategy (ibid.). While doing the redesign, removing some old parts completely should be considered. The more backward compatibility is retained, the higher the performance limitations are.

Since the dynamic landscape is continually changing, it is not wise to keep the product unchanged, particularly when the initial solutions are rapidly put together to fulfil only some functional considerations. Redesigning the old radio network features could improve the system usability of the system, but so far, the network suppliers have not been doing this. Obviously, the main reason is the lack of resources due to the present laborious product development. Another reason could be that the customers are not expected to be willing to pay for later improvements. In addition, the significance of system usability has not been completely recognised yet. As described in the theoretical framework, when developing complex systems, nature occasionally optimises the designs and reduces the complexity. It would also be a worthy measure for the technology developers.

8.3 Coevolution related guidelines

8.3.1 Empathic feature research

Feature research, in this context, means getting ideas for new features from the customers. Presently the 'research' is done so that a few persons from the supplier company discuss with the customers a few times a year in meetings. As discussed earlier, this kind of knowledge acquisition is not a very efficient way to start the new feature development. Better insight can be achieved by using ethnographic methodology, which is pointed towards seeing the world from the point of view of participants (Crabtree/Nichols/O'Brien/Rouncefield/Twidale 2000[185]). Ethnography is a naturalistic method in that it relies upon material drawn from the firsthand experience of a fieldworker in some setting and seeks to present a portrait of life as seen and understood by those who live and work within the domain concerned (ibid.). The goal of ethnographic methods has always been to see things from the point of view of a person dramatically different from the researcher, and to then describe that point of view in a language that can be understood by those from the researcher's own culture. Therefore, the value of an ethnographer to a product-development team should be obvious. That is, the product development team must see the customer's point of view and be able to translate that point of view into product characteristics (Wilcox 2001b[186]).

An *empathic research* method allows good understanding of the operating problems and provides a strong starting point for defining the solutions. In this method, the researchers spend a rather long time with the operators by observing and interviewing the users. As Alvesson/Sköldberg 2000[4] put it: Understanding calls for living (thinking, feeling) oneself into the situation of the actor. In this way, the researchers get a realistic image of the operations and are able to create initial specifications that are closer to the required solutions. The word 'required' in this context does not exactly mean what the users would answer, if they were asked to specify the new system. Of course, the users' opinions must be taken into consideration, but the basic idea is to learn the problems and then create comprehensive solutions to eliminate the problems. This kind of approach allows a higher level of thinking in solution design – a kind of metasystem transition so to speak (refer to chapter 8.1.2 'Metasystem transitions').

Unlike traditional development projects, which rely on periodic bursts of input on users' needs, projects in turbulent business environments require *continual feedback* (Iansiti/MacCormack 1997[182]). This feedback is required as a selective force guiding the product development. To acquire and use this valuable information, the development process must be able to:

- Get a rich understanding of customer's problems (continual *empathic research* with customers)
- Test alternative technical solutions (e.g. via the Internet and multimedia)
- Integrate the knowledge gained of both markets and technologies into a coherent product (ibid.)

To allow rich input for new features, there could be a small team of researching designers doing research with the customers by:

- *Observing* them encountering problems
- *Doing interviews*
- Seeing *the operator-designed applications* they have had to produce

This approach would establish strong coevolutionary links with the operators already in the very beginning of the development cycle. After this, the solutions would be defined based on the things learned in the research and a vision of a product with high added value.

8.3.2 Operating task research

One of the results of this research was that the worlds and problems of network operators and suppliers are quite different. In addition, various operators have different ways of doing things. The suppliers have also kept the management systems rather generic, because they know that their own management systems are not the only ones used due to the multivendor environment. How then, could the suppliers develop usable networks? This requires enhancing the coevolution between suppliers and operators. The suppliers could contribute to this by doing research with operators to learn their ways of working. Ethnographic methods are very suitable for this kind of research, as ethnography aims at developing a thorough understanding of current work practices (Simonsen/Kensing 1997[7]).

The results of the operator task research would enable the supplier to turn the disadvantage of modest tool suitability (generic systems) to advantage by developing best practices for operating tasks and adapting the system to better support the tasks. However, this does not mean that the ideal would be a strictly defined rigid system. Flexibility to accommodate variation in operating is still required. There would be two immediate benefits for the approach (see also Figure 42).

- The number of different ways of operating would reduce and hence it would be easier satisfy the operators, when their requirements are not quite so different.
- The task oriented management applications would lead to good system usability.

An alternative possibility would be to mass customise the products, so that the customers could use the graphical tools of the supplier via the Internet to build specifications for the wanted functionality. This would require very modular products, where the set of functionalities can be built from modules. This possibility is not studied any further in this document.

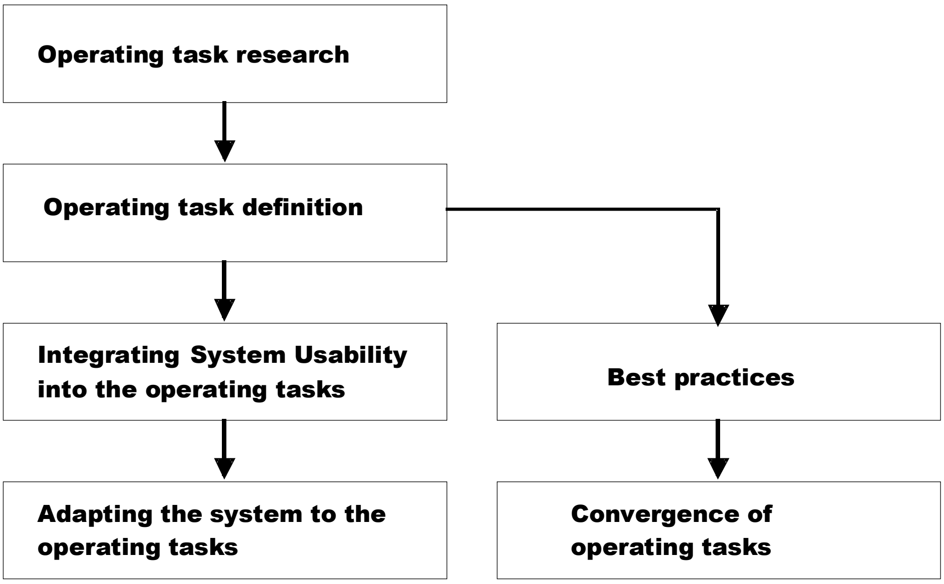


Figure 42. Operator task definition and utilisation. System usability is integrated in the operator tasks and the system is designed accordingly. By adapting the system to the transcendent operating tasks, the system usability could be improved.

We can investigate the operator task design with the aid of an example task. Refer to the Figure 34 in the case descriptions displaying the BTS commissioning process of a certain operator. The example process is quite complicated requiring several different work groups and co-ordination between the groups. The distinctive feature in the figure is that the groups cannot perform their tasks independently from each other, but instead occasionally have to synchronise with another group, and wait until this other group has accomplished some intermediary procedure before they can continue with their work. This interdependence requires co-ordination and extra effort, thus undermining the efficiency of the process.

The redesigned process (see Figure 35) is simplified substantially and the interdependencies are removed, so that all the different work groups can perform their tasks without synchronisation with the others. Designing efficient tasks might also trigger modification needs in the system, as the system can be quite inflexible. That is why system alterations are required in connection with the design of recommended processes. This means that we have, in addition to coevolution of operators and suppliers, also coevolution of the system and the operating tasks. However, the adaptation to the best practices does not remove the need for flexibility in the system. Other ways of working should still be possible.

8.3.3 Release updates and time pacing

Presently, the SW-releases of the different subsystems come out at different times without a proper synchronisation. The radio network functionality (BSS software release) is always delivered before the managing software release (NMS). Sometimes, the management support does not come even in the very next NMS release, but even later. This causes extra effort to the operators, as they have to manage with the new BSS version without any NMS-support for the new features. In practice, this means that the operators design huge macros and other tools to be able to use the new functionality. It is awkward to the customers, if they have to worry about the contents, release times and compatibility of different parts of the system.

Synchronisation requires closer co-operation between the product lines taking care of the different subsystem releases (Cusumano/Yoffie 1998[187]). This means strong coevolution *within* the company. One way of improving the early coevolution within the company would be to utilise the animated multimedia descriptions of the new BSS features (see chapter 8.2.1 'Product development'). The practical challenge here is how to test a subsystem that requires another subsystem to work, before this other subsystem is completed. Perhaps an interface simulator could be the answer for this. In that case, the NMS software package could be tested with the BSS-interface simulator, which is updated in each release before finalising the BSS software. Anyway, this solution is not based on the theoretical framework of this research, but on the existing product development paradigm.

A possibility following the theoretical framework, would be an adjoint development, so that each BSS function and its NMS-support applications are created and tested together. The first requirement for this would be to make the BSS software modular (refer to chapter 8.1.4 'Modular software structure and hierarchical architecture'). Then the respective functionalities of BSS and NMS would be created together, but instead of one giant package, the software would consist of numerous small entities. When designing the entities, as much as possible of the combined functionality would be integrated in the BSS elements (refer to chapter 3.1.1 'Hierarchy, modularity and autonomy'). The functionality would either be totally autonomous, or activated from NMS to perform certain tasks and provide information to the operators in the OMC. This approach demands distributed intelligence in the network elements instead of the present concentrated control (BSC, NMS), since the low level subunits cannot be autonomous without some intelligence. As the software would consist of many small packages, these packages could be loaded little by little thus avoiding massive update processes. It would even be possible to update the software directly to the network elements via the Internet. This would also enable continuous development and component delivery instead of delivering a large package once a year.

In addition to the subsystem release deliveries, there should a proper time pacing of the products. Time pacing means launching a new product or service e.g. every six months, rather than whenever the competitive response is needed or whenever the product is finalised. In the example company (Hi-Tech Inc.), there is no accurate time pacing in the series of SW-releases. The releases come out approximately once a year, but not at the same time of the year every time.

Pace decision is strategic. Brown and Eisenhardt (Brown/Eisenhardt 1998[184], Eisenhardt 1999[188]) discuss the importance of time pacing. The releases do not necessarily have to be more frequent, but regular. If the releases are properly paced with time, this creates a rhythm with which the company can synchronise many other things. Rhythm helps to orchestrate the complicated and error-prone processes involving many people and resources in the developing company. Time pacing enhances strategic option as the managers can use the periodic transitions to survey the competitors. It also creates predictability in the eyes of the customers, as they know exactly when the next release is coming out (ibid.). Hence, time pacing serves many functions with several players in the coevolution. In addition, the contents of the coming releases should be shared with the customers early enough, so that they can prepare their activities based on the future products. When operating in a dynamic landscape, it is good to know at least some things in advance.

8.3.4 Links between designers and users

A mismatch between the views of the designers and the expectations of the users of computer systems has been cited as one factor that contributes to lower levels of system usability (Kadoda 2000[189]). The research by Kadoda showed that the designers' images of an actual user were significantly different from the users' self image.

Madsen/Borgholm 1999[190] and Buur/Bagger 1999[191] also call for more active collaboration between users and designers, for example in the form of participatory workshops. Coevolution does not work properly, if the coevolutionary links between the agents are too weak. A good example of this is the relationship between the network planning tool designers and network planners. Originally these two groups were geographically together, but later they were separated into different places[a]. This weakens the coevolutionary links between these groups, and there is evidence that the outcome is not very good. Particularly in the case, where the knowledge required for the development is available within the same company, the knowledge should be utilised to the full extent by establishing a strong coevolution between the units. This kind of complex knowledge cannot be transferred via weak communication links (Hansen 1999[192], Cliffe 1998[193]). Actually, the situation was later changed so that the tool development is now done by an outside company. This situation is even more risky, and a lot of care should be taken to keep the coevolutionary links strong. It requires a lot of face-to-face discussions and hands-on co-operation instead of telephone calls and emails.

Based on this research, the same phenomenon seems to be partly responsible for the modest level of the network usability. Many times the products are designed, built and tested before they are shown to the customers. At that stage, it is too late to make any radical changes to the product. The design team must have face-to-face discussions with representative users, rather than relying on third parties for market research (Meyer/Seliger 1998[194]). Better yet is to use the ethnographic approach and spend longer periods of time with the operators. The coevolution can then be built on a) empathic feature research, b) operating tasks research, c) a flexible product development process capable of using continual feedback during the development cycle. Then, we just have to provide the feedback. This requires a lot of co-operation with the users and all conceivable methods to provoke useful comments. For example, the mentioned audiovisual multimedia is one way.

However, nothing can replace the real world experience in the real context, and this should be an integral part of the co-operation. Good product development requires strong coevolution between

---

[a] This is based on author's own experience in the company.

the users and the designers, as this research once again shows. That is particularly true in the case of complex technical products, where the solution space is so vast that there are millions of ways you can go wrong.

8.4 Challenges and solutions

In this chapter, we shall list the solutions connected with the challenges found earlier. Here, the categorisation is done based on the solutions, not based on the challenges as before. The solutions have been divided in three different groups dealing with: complexity, dynamic landscape and coevolution. There might also be some weaker connections in addition to those marked in the figures, but at least the shown connections can be identified rather easily.

Table 13. Challenges, in which the complexity based solutions would be helpful.

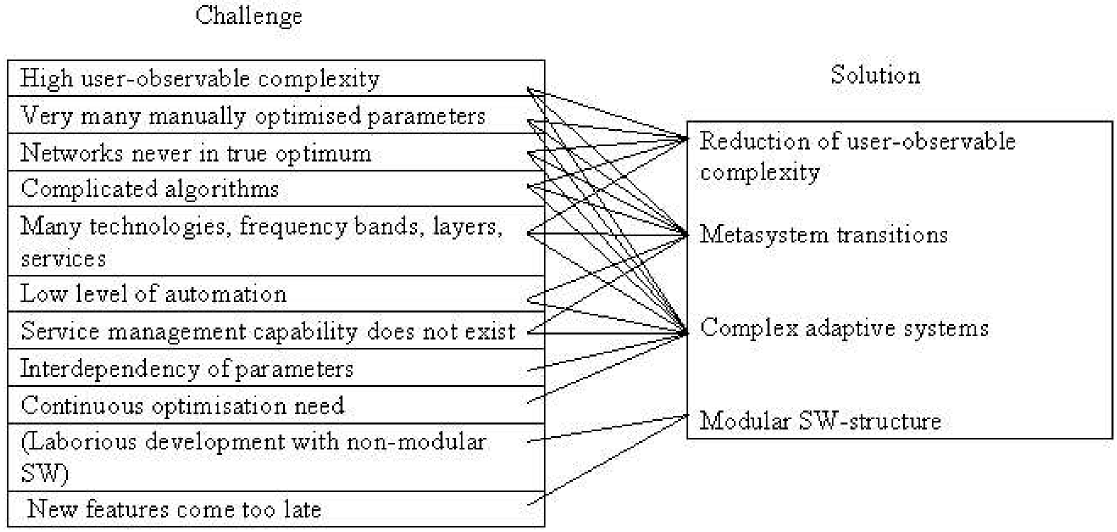


Table 14. Challenges, in which the dynamic landscape related solutions would be helpful.

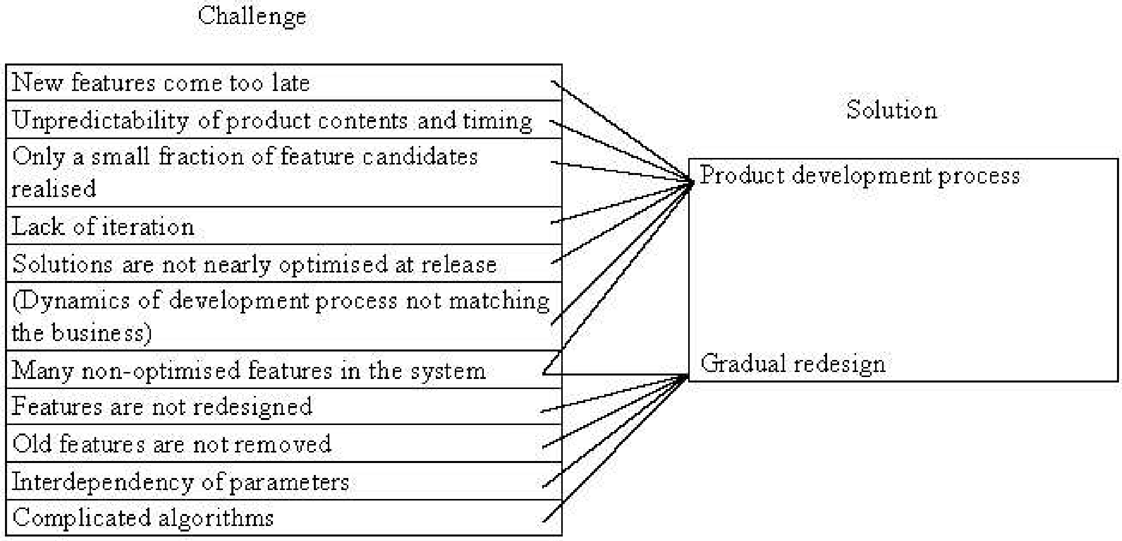
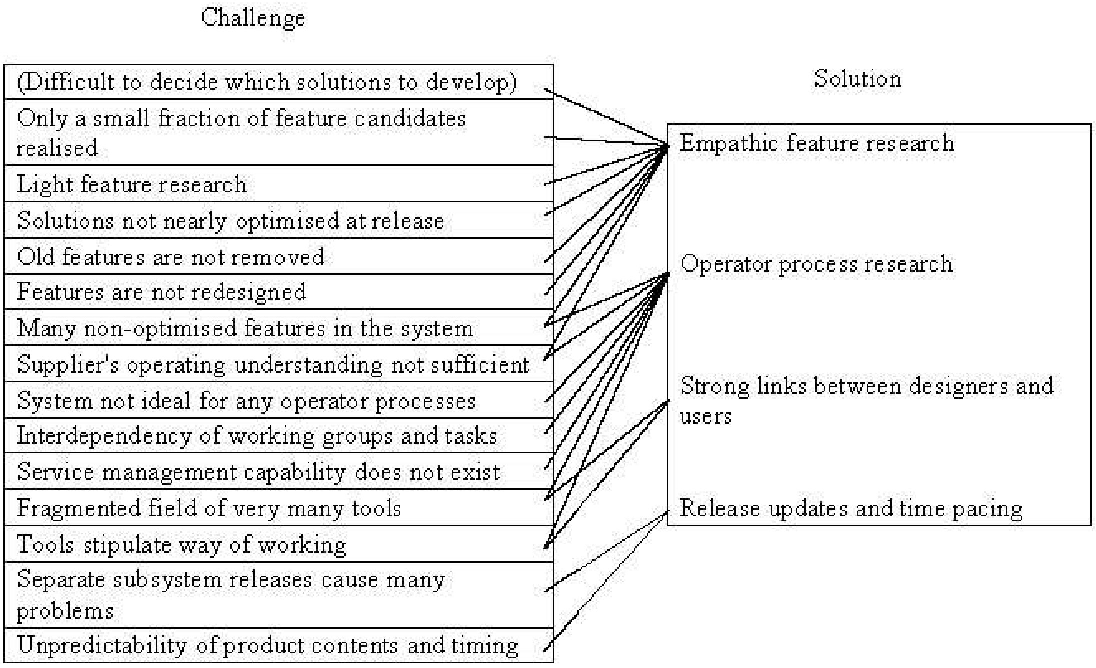


Table 15. Challenges, in which the coevolution based solutions would be helpful.



9. SUMMARY

In this chapter, we summarise the main contributions of the research. We shall briefly address the relation of the research questions and the results in the first sub chapter. Then in the three sub chapters following this, we shall review the contributions. In the second sub chapter, we introduce the definition for system usability, which can be used for measuring and comparing system usability of different complex technical systems. In the third sub chapter, we list some guidelines, which can be used for achieving better system usability. Finally in the fourth sub chapter, we characterise the coevolutionary nature of operation of different players, mainly network suppliers and operators.

9.1 Research questions and results

In *the first research question*, we asked how would it be possible to understand system usability of a technical system via its complexity. This question was answered by creating the theoretical framework. Since usability implies a relationship of a user (or various groups of users) with the system, it is quite reasonable to talk about user-observable complexity. As we recall from chapter 4.3 'The relation of user-observable complexity and system usability', both the system itself and also the user interface contain a certain degree of user-observable complexity. We have divided the user-observable complexity into two components, namely effective and random complexity – the former being the useful part for the user and the latter being a parasite component. Complexity has also other kinds of effects on system usability besides the increased user-observable complexity. For example, if the various parts of the system have a high interdependency (K-factor), it means that handling of such a system becomes very difficult. This means that the excellent solutions fade away due to conflicting constraints, thus making it impossible to reach good system usability.

In *the second research question*, we asked how to improve system usability by managing complexity. The answer to this question also was addressed with the creation of the theoretical framework. In the process, we found many ways of improving system usability. For example, it is wise to keep the number of cross-connections (K-factor) relatively low and also build the system by using hierarchical architecture making use of autonomous levels. The system usability can also be improved by metasystem transitions, when the complexity gets too high for the users. It is also useful to minimise random complexity and reduce effective complexity of the system by radical innovations. These are some examples of complexity management in order to improve system usability. A complete list of the discovered means can be found in Table 5.

In *the third research question*, we asked three different things. 1) Are cellular radio networks *complex* technical systems? Based on the simulations, observed empirical phenomena, comments of the informants, research literature, as well as the remarks of the industrial experts in the validation phase, we can state that these systems are complex technical systems.

2) What is the present system usability of cellular networks? Based on the empirical research, we can state that the system usability of these systems is currently rather modest. The main reasons for this situation could be a) The present user-interface oriented usability paradigm does not address the inherent complexity of the systems, b) Usability issues have not yet thoroughly conquered the relatively young industry, c) Complex technical systems are a very demanding domain for (system) usability research.

3) In case the system usability is not good, can it be improved by managing complexity? Based on the theories of natural sciences, it seems that nature is managing complexity when producing complex systems. For example, the K-factor seems to self-guide to a rather narrow zone between 0 and N-1. Nature also uses platform development (tinkering) and modular architecture, where different layers have a rather high autonomy. It seems reasonable that similar actions would be useful also in man-made complex systems. In this research, we have created a set of guidelines for improving the system usability of complex technical systems (see chapter 8 'Using the theoretical framework to find guidelines'). Although we have to note that this research did not yet produce conclusive evidence for the effectiveness of these guidelines in technology, based on the comments of the industrial experts there is reason to believe that they would be useful in the technical domain.

9.2 System usability

System usability is defined with the aid of the effort operators have to put in operating the system. The total effort in system usability is divided into five different categories:

- Planning (E_{Plan})
- Building (E_{Build})
- Using (E_{Use})
- Improving ($E_{Improve}$)
- Managing the services ($E_{ServMan}$)

All the operating efforts can be listed under these categories. System usability is the reciprocal of the sum of the various efforts (see Equation 7).

$$SystemUsability = \frac{1}{E_{Plan} + E_{Build} + E_{Use} + E_{Improve} + E_{ServMan}} \quad \text{Equation 7}$$

where E_x = Effort used by operator

The different efforts can be measured, for example, with the aid of man-hours the different user groups have to invest in their tasks, amount of training that is needed to handle the necessary equipment, employee turnover rate giving an indication of how tedious the work is considered to be etc. To follow the main theme in the theoretical framework, we can say that the selective force for the system development comes from the effort used for operating the system. Searching for system usability of a complex technical system can be considered as an effort-minimising task, where the objective is to find the deepest valleys.

We can compensate for the unequal significance of the different efforts by adding weight factors for the efforts (see Equation 8). In addition to the weight factors, system usability can be calibrated with the system size (see Equation 9). A *relative* system usability is the absolute system usability divided by the number of features in the system (see Equation 10). This enables us to get an implication of how the number of features affects system usability. For example, increasing the number of interdependent features could decrease the relative system usability and eventually lead to a complexity catastrophe. In addition, if we want to compare the system usability of several completely different complex technical systems, we have to introduce one more weight factor for the system usability of every system (see Equation 11).

$$SystemUsability = \frac{1}{\alpha E_{Plan} + \beta E_{Build} + \gamma E_{Use} + \delta E_{Improve} + \varepsilon E_{ServMan}} \quad \text{Equation 8}$$

$$SystemUsability = \frac{(Nr)_{BTSs}}{\alpha E_{Plan} + \beta E_{Build} + \gamma E_{Use} + \delta E_{Improve} + \varepsilon E_{ServMan}} \quad \text{Equation 9}$$

$$SystemUsability = \frac{(Nr)_{BTSs}}{(\alpha E_{Plan} + \beta E_{Build} + \gamma E_{Use} + \delta E_{Improve} + \varepsilon E_{ServMan}) * (Nr)_{feat}} \quad \text{Equation 10}$$

$$SystemUsability = \frac{\phi (Nr)_{BTSs}}{(\alpha E_{Plan} + \beta E_{Build} + \chi E_{Use} + \delta E_{Improve} + \varepsilon E_{ServMan}) * (Nr)_{func}} \quad \text{Equation 11}$$

System usability is also affected by operating task procedures. It does not pay to improve system usability without taking the operating task procedures into consideration. Both the system usability and operating tasks should be considered in connection with each other.

9.3 Guidelines for achieving system usability

It is recognised that there is also the organisational complexity component contributing to the success of a company, but since the focus of this research is not on organisation science, we did not go deep in that subject. However, based on other research (McKelvey 1998[139], Maguire/McKelvey 1999[195], Brown/Eisenhardt 1998[184]), it seems that both the firm's internal and external complexity are important factors and should not be ignored in order to avoid the complexity catastrophe.

In this research, we have found some guidelines for improving system usability of complex technical systems. In the following, there is a summary of those guidelines compiled from the more detailed descriptions in chapter 8.

Complexity related

1. User-observable complexity reduction

As complex adaptive systems, humans for example, tend to identify regularities (effective complexity) from the available data and ignore the random complexity while observing the environment, it would be useful to design the complex systems to assist the users in this tendency. This would mean eradicating the random complexity and occasionally reducing the effective complexity.

- Removing random complexity
- Reducing effective complexity by major innovations

2. Metasystem transitions

As the complexity of technical systems tends to increase, sometimes it would be useful to make a metasystem transition by introducing a new higher level, through which the users access the system, and hide the more detailed levels below. This method also reduces user-observable complexity.

3. Complex adaptive systems

One good way of hiding the low level complexity is using complex adaptive systems. These systems have the capability of creating new information and thus adapting to the dynamic situation without a runtime manual selection of pre-stored possibilities (parameter tuning).

4. Modular structure (K-factor reduction) and hierarchical architecture (complexity management)

There is an imminent danger of complexity catastrophe, if the number of connections in a complex system is excessive. Modular structure is a good way of preventing the number of direct links to other parts of the system from rising too high. To manage a high level of complexity in a system, it is also useful to create a hierarchical structure, where every level and subunit has a high degree of autonomy. This eliminates the need for a very high number of direct links to the common control point and very high processing power in the central point.

Dynamic landscape related

5. Product development process/business dynamics matching

The product development process should provide a product modification rate, which is suitable for the business dynamics. For example, if the dynamic business landscape changes faster than the products can be modified, it is not possible to keep the products on the fitness hills of the landscape.

6. Gradual redesign of old features

As the features are usually quickly designed from engineering point of view to offer the required functionality, it would be useful to improve them in later releases after the initial introduction. This would prevent the system from becoming a collection of obsolete, low usability solutions.

Coevolution related

7. Empathic feature research

To increase the understanding of the operator world on the supplier side, as well as give a better starting point for new feature development, it would be good to use the empathic feature research method. In this method, the designers spend a longer time with the operating personnel to learn the problems, so that they can then create the solutions for eliminating the problems. Asking for feature candidates is not a sufficient starting point for creating excellent products.

8. Operating task research => recommended best practices + system adaptation

To further increase the operating task understanding, the suppliers should do operating task research. This understanding could be used for creating recommended ways of performing the tasks. If the system is adapted to fully support these tasks, the synergies between the good processes and the system would further amplify the benefits. However, there is still need for flexibility in the system, so that other ways of doing and also future improvements are accommodated.

9. Release updates and time pacing

The releases required for operating should be delivered together. This means synchronisation of the delivery of different subsystem functionality, so that the customers do not have to create work-around solutions in order to cope in their work. The deliveries should also be time paced, so that they come out regularly at the same time of the year.

10. Strong links between tool development and users

Particularly the designers of complex technical systems should have strong links to the users in order to maintain the touch of reality. The solution space in complex systems is huge, and if the solutions are created without a strong guiding force, it is very likely that the designers get lost and the outcome will not meet the practical requirements.

9.4 Coevolution of suppliers and operators

The concept of coevolution has been applied before in the business world (e.g. McKelvey 1998[155]). However, in this research the coevolution is applied in product development of complex technical products. Coevolution can be applied to many levels, for example, inside a company there are many development units living in coevolution with each other. The main emphasis in this research, however, is on the company level coevolution, where the technology suppliers, their customers and some other forces mould each others adaptive fitness landscapes by their actions.

The supplier companies create complex technical systems, which they offer to the customers on the markets. Each of the companies tries to reach as high as possible in its own fitness landscape. Since the companies are interconnected by coevolutionary links, they change each others' adaptive landscapes, as they make their adaptive moves (refer to Figure 20). In this way, this coevolution makes the landscapes of all players dynamic. Consequently, they all have to survive in a dynamic landscape by improving their products with the aid of customer feedback as a selective (guiding) force.

For example, when a supplier creates a new technical system, which is a combination of certain, available technical components, this act places the product to a certain place in the solution space (technical landscape). The product has a certain fitness represented by its height in the landscape. The aim of the company is to take the product to the highest hilltops in the landscape to maximise its fitness. This would be much easier if the landscape were static, since there would be plenty of time to get there, providing that a suitable hill has been found. However, in a coevolutionary situation, where the landscape is always dynamic, the task becomes much more difficult. This is particularly true in the case of complex hi-tech products, where the landscape is said to be 'turbulent'.

Hence, if supplier Y comes up with a good idea for a product, this will change the landscape of supplier X and the hill under their product may vanish. This means that they have to *innovate* new products again in order to find a new fitness hill in the solution space and climb up there, which in turn moulds the landscapes of the other connected players. The suppliers may also *gradually* improve their products and, in this way, slowly climb upwards in the landscape. Typically, a new good product lowers the fitness of the competing suppliers (by moulding their landscapes via the coevolutionary links) (Kauffman 1992[126], 1995[134]) and improves the fitness of the operators using this system. The operators are also linked to each other (as well as to the suppliers) by the coevolutionary links and affect each others' landscapes when striving for better fitness in their operating business. The whole metasystem consists of several system suppliers and operators, as well as some external players, such as standardisation institutes and legislators.

There are also certain dangers involved. The first one is **A**) *Error catastrophe.* If the modification pace of the product is too slow (e.g. due to long and rigid product development process), the products cannot track the dynamic landscape and the fitness hills escape (This type of error catastrophe is not possible if the landscape is static). Or if the modification pace is too fast compared to the steering force (feedback from the users of the product), the products flow down along the hill slope. The second danger is **B**) *Complexity catastrophe.* If there is a heavy cross-coupling (too many interdependencies) between the multiple technologies or components forming the complex technical product, the fitness hills fade away due to conflicting constraints, as all solutions become compromises (Kauffman 1992[126], Fleming/Sorenson 1999[129]).

In complex technical products, the solution space is enormous due to all the possible combinations of the many parts. Thus, there are always excellent solutions to all conceivable problems, but the difficulty is that those good solutions are very difficult to find – for the same reason, vastness of the solution space.

10. VALIDATION

10.1 An assessment by industrial experts

As we already mentioned in the methodology chapter, the natural sciences are inclined towards seeking the 'ultimate' truth. What can we do to get an idea of how well we have succeeded in this? For this purpose the investigator may wish to arrange a session to which are invited knowledgeable individuals from each of the several interested source groups (Lincoln/Guba 1985[197]). Lincoln/Guba suggest a technique called 'member checks' whereby data, analytic categories, interpretations and conclusions are tested with members of the stake holding groups. Applying this suggestion, we performed an *industrial expert panel validation*.

In the research, we also found out that the worlds of operators and suppliers are quite different, but in addition it is also recognised that the academic and industrial worlds are somewhat different. The validation with the industrial experts also served as an *experiment* to see how readily the research results can be understood and exploited in industry. The process also illuminated the supplier's side of things, as most of the participants were from Hi-Tech Inc. The group consisted of five managers of Hi-Tech Inc., one manager from an operator and one senior researcher from an independent research institute, who has experience of other complex technical systems besides cellular networks.

The review group did not get the whole text, but only about 25 pages containing a) short description of the conventional usability paradigm, b) the basic idea of system usability including the mathematical definition, c) the high level interpretations of the results d) the guidelines to improve system usability, and e) the conceptual model of coevolution in product development. The whole text was not given to see how well the results are understood without the theoretical framework. It was also anticipated that the participants are too busy with their work to read such a long and theoretical document. In addition to the text, also a list of questions was delivered to the partakers. The list was in the form of arguments concerning the text, and the requested action was to comment the arguments.

As we mentioned in chapter 7.4, if no current product exists having the new features even in the most primitive form, the users have no foundation on which to formulate their opinions (Leonard/Rayport 1997[169]). This was very true also in the validation, as the guidelines are quite theoretical and not applied into any engineering solutions yet. None of the participants commented all parts and some of the requested partakers did not give their comments at all, which also indicates that the task was not easy. Particularly, the guidelines for improving the system usability were very difficult for most participants.

In the following, we have a summary of the comments compiled under five different headings that follow the theme of the research. Most of the participants gave their comments in writing using very concise language. Since the answers were so short, the author had to put some more words to the stories to make them more readable. Still, the author's part is very small in the following recitation.

Complexity

A manager in Hi-Tech Inc.'s product development tells that all operators have a workflow process, which links the tasks of various working groups together. There is quite a lot of interdependence in the tasks of the different groups. The networks also represent high user-observable complexity. As the product development manager put it: "At least we have made it [the network] complex". He continues by telling that vendors typically cannot tell how different things affect each other. This means that operators have to create this understanding by doing practical experiments. It could be useful coevolution to co-operate with operators in this knowledge creation – it would be a mutual benefit. He also describes the environment by saying: "The dynamic landscape canNOT be highlighted too much. Subscriber behaviour and environment changes all the time". Since many things are inter-related, it seem impossible to get everything in top condition. As the manager phrases it: "One part is [in] optimum, some other is not. Optimisation is [an] endless project". This also tells about the complexity of the system.

There are many technologies, frequency bands and layers in the radio networks. What makes things even more diverse is that there are always equipment from several vendors, as one manager in Hi-Tech Inc. added. The reviewers agree that complexity is increasing every time when new technologies are introduced. From that point of view, reduction of user-observable complexity would be very useful. A product development manager of Hi-Tech Inc. made the comment that reduction is not the only issue. In his opinion, reduction often means less functionality, so it would be useful also to hide the complexity. Although he had not completely understood the concept of user-observable complexity reduction, his comment validates the basic idea of both the user-observable complexity and metasystem transitions. As user-observable complexity consists of random complexity and effective complexity, the reduction of the former does not reduce functionality, and the latter one makes use of radical innovations to reduce effective complexity by still maintaining the functionality, whereas the metasystem transitions hide the low level complexity whatever it is. Since the validators did not see the theoretical framework, the comment is independent from that. Metasystem transitions were considered necessary by all reviewers. As the product development manager remarked, once in a while the lower levels must be reviewed for changes. Modular software and hierarchical structure were considered definitely a useful idea by many validators. A product development manager of Hi-Tech Inc. confirmed that this principle is very much used in the coming management systems.

Different worlds

The reviewers agreed that there are very many different applications for managing the system, different technologies also have their own management applications. A product development manager of Hi-Tech Inc. tells that the different worlds are even more visible with independent OSS application vendors. They usually have very generic applications for the operator tasks. The management systems of network supplier companies also have fragmented sets of tools. According to him, there might not be any tool to support certain phases in the processes, in which case the operator must either create their own tools or search for 3rd party tools. Many times, the tools are quite inflexible and allow only one way of performing the tasks. A product development manager of Hi-Tech Inc. remarked that tools actually should stipulate the way of working and in this way guide the user through the tasks. In principle, this is a sound idea. However, the supplier has not succeeded in designing the tools to support the tasks in such an efficient way that the users would like to follow it without exception.

Many of the reviewers confirmed that the solutions are not optimal for any particular operator, because requirements are so different. It is not possible to please everyone. One Hi-Tech Inc.'s product development manager tells that supplier's understanding of network operating is not very profound, and typically it is limited to some special area. He added that in the product development departments particularly, the practical understanding is too modest. The reason is that the designers are not in very close connection with the operating people. As another manager in Hi-Tech Inc. said, most designers never see the operating processes in practice. Designer's scope is also very limited – one designer deals only with a small part of the processes.

NMS applications have not been entirely relevant for operator environment, for example the functionality for mass operations has been missing, as one manager in Hi-Tech Inc. noted. The reviewers agree that the level of automation has been rather low. All reviewers verified that operators would like to minimise manual effort and save expenses, so in that sense the level of automation should be much higher. However, some of the reviewers are a little sceptic towards automation in complex systems. One Hi-Tech Inc.'s product development manager remarked that in complex systems, some automatic change in one place could cause nasty effects elsewhere. On the other hand, he points out that presently there is not enough system understanding in the supplier's organisation. More system level research and complexity understanding is required, if these things are to be solved.

Service management does not exist today. The reason is that network management and operator business management connection is not well understood yet. As a manager in Hi-Tech Inc. expressed the situation: "In operator world link between network management and business management not necessar[il]y exists". Operators or suppliers do not know how these things should be connected.

Product development

Only a small fraction of feature candidates are ever realised, some are ranked out for business reasons, some due to lack of resources. A product development manager in Hi-Tech Inc. tells that product development programs are already full for several coming releases. He also agrees with the notion that solutions are often not exactly what the customers want and many times they also come too late. The reason is two-fold. a) The development cycles are too long, b) The growing understanding cannot be exploited within the development cycle, since iteration is not possible. The manager regrets that nobody can tell several years ahead what will be needed at the time when the products would be available. According to him, the landscape is too dynamic for that. "The product development seems to be like shooting in a moving target". The product development process should be able to cope with the dynamic landscape. He tells that some product lines of Hi-Tech Inc. have made a few iteration experiments in product development, but in general no iteration within a development cycle is currently used.

Lack of redesign is aging the systems. A product development manager of Hi-Tech Inc. stated that redesign in OSS is very slow, the 2nd generation management systems are now in hand after ten years. In BSS features there is no redesign, but the new features are added on top of the old system. He also tells that features are not removed either. "We do not know if someone is using this"-syndrome seems to prevail. For example, there are at least 100 statistic counters in the system that are not relevant any more, but are still there. Another manager of Hi-Tech Inc. put his concern in words by saying that the higher the backward compatibility, the higher the performance restrictions. According to the product development manager, gradual redesign is a must and also bigger restructurisations (metasystem transitions) are needed.

The product development manager continues by telling that empathic feature research has been tried in a project and the feedback has been good. However, in his opinion the customers should be selected carefully for this kind of research, as the influence of an individual operator can be very strong. It also places high requirements for product management. According to the same manager, operator process research is a good idea in principle, but if too many details are considered, the outcome tends to fill one operator's dream and leave the rest unsatisfied. However, this is a question of how the results are used. In this work we proposed that the results would be used for designing recommended operating task procedures and adapting the system to these procedures. The independent complex system researcher tells that feature research is closely linked with operator process research, since operating understanding is required as a basis for the feature research. In her opinion, both of them must be considered in the product development process. According to her, there should be some flexibility in the operating processes and also in the system. She also tells that there seems to be very little operating process research literature available.

According to a manager in Hi-Tech Inc., synchronisation and time pacing of subsystem releases is a must on the markets – the system is as weak as the weakest subsystem. Nevertheless, system level planning does not exist today and this is causing a lot of extra effort, he tells. As the operator representative pointed out, also the lack of documentation prior to the release is a problem. The operators cannot prepare for the releases. The Hi-Tech Inc.'s manager tells that strong links between designers and users have also been tried in a project and it has been found to be mutually motivating. However, in his opinion care should be taken that one user cannot dominate too much, so that the end result would not fit others. In the current situation, the supplier must navigate through different requirements.

System usability

Usability design has been very limited in cellular networks (and system usability non-existent). A product development manager in Hi-Tech Inc. tells that in OSS the *traditional* usability paradigm has been in use for several years, but in the radio network there has not been any usability design on BSS level or network level – only some individual efforts on network element level. According to him, it would definitely be useful from system usability point of view to design good operating procedures and adapt the system to support those procedures. In his opinion, this should be done when designing the network functionality, not afterwards.

According to the independent complex system researcher, system usability is a novel approach to usability and quite likely to lead to better usability in complex systems. In her opinion, the research contains new kind of empirical material, which could be interesting for many people. She also thinks that the findings seem quite credible, since similar problems have been observed in other complex systems too. She has discovered that the basic problem usually is that the equipment supplier does not sufficiently know the operator's world and needs. It was also pointed out that usually automation increases uncertainty. Nevertheless, at least one operator, which uses NMS very much in their daily work, commented that in their environment they trust it blindly. It seems to be a question of getting used to the idea – those people that are accustomed to using MML do not trust the graphical tools, as they give so little feedback.

A manager in Hi-Tech Inc. pointed out that measuring network operating effort directly based on man-hours might be inaccurate, since the skills of various people are different. Some experienced expert can do more in an hour than another person in a day. This is recognised and should be investigated in further research.

Improvement suggestions (guidelines)

The following rather longish text commenting the suggested guidelines is based on an interview with a manager of Hi-Tech Inc., who was familiar with the theoretical framework and also with the processes and challenges in the company.

Random complexity occurs due to misguided design, implementation and commissioning. All these phases create unintended phenomena not improving the end result. If all these unwanted traits were removed, the system usability would quite probably improve. Recently, the user-observable complexity has been reduced, for example, by creating common radio resource management, where several different radio networks can be controlled like they were only one network. This is an example of a metasystem transition. Reducing user-observable complexity by metasystem transitions is a relevant means to improve system usability in practice.

It has been noted that the interdependency of different parts of the system has become very high. This has lead to a situation, where touching various parts of the system is being avoided, since all the consequences are not known. This also indicates that there would be a need for a restructurisation.

Constant manual optimisation of the parameters in a complex network, where all interdependencies are not known, is laborious. Adaptive systems would be, in principle, a good way of eliminating the manual optimisation. However, there is a challenge in accomplishing it without a risk for mistuning the network. The idea seems so distant that it is difficult to see what the realisation would be like. Earlier there were some difficulties, for example, in the CDMA power control, but those problems have been solved. There are always complications with new kind of solutions in the beginning, but this should not be able to prevent the progress.

It has also been noticed that the present way of always adding the new features on top of the old ones without any redesign, leads to an impossible situation in the long run. According to the interviewee, it is likely that after a few more releases a limit is reached after which the increasing complexity eats up the intended benefit, if the present way of development is continued.

Modular architecture is a good principle of building complex technical systems, but the result always depends on the quality of implementation. Particularly, if the development of the existing monolithic system is continued with modular design without restructuring the old part, the outcome cannot be satisfactory. A proper restructurisation is needed at this stage. Also after the restructurisation, continual redesign is needed to keep the system up-to-date. However, the modular structure would make it easier, as the modules can be replaced one by one. Some experience of the hierarchical structure already exists, but it this structure is not in general use yet. Traditionally, the fault management has been implemented so that the network elements send all the raw fault alarms to the NMS. There the situation is handled by using various filters to reduce the number of alarms, since it is impossible to check all of them. In practice, the operators many times ignore the alarms completely and wait for complaints from the end users. It would be better to design the architecture so that the network elements would be as autonomous as possible to avoid straining the management centre. This direction has been assumed already in the base station development. Hierarchical layers seem to be useful in building complex technical systems.

The mismatch of the present product development process and the dynamic landscape is a clear problem, which has been noted to some degree. Obviously, the innovations in the product development process have so far concentrated on improving the product creation as an internal process. Taking into account the changes in the external world during the product cycle, has not been properly considered. The only alteration that is possible in the product during the cycle, is removing some parts from it. In practice, the changes in the operating world during the 2-3 year long development process do not show in the end result. This is a serious problem, since the operators cannot tell what they want 2-3 years ahead. Later, when this information is available, the products cannot be changed anymore due to the inflexible development process. This can lead to a situation, where unfit and obsolete products are produced. The only way to improve the situation is to let the customers follow the development and utilise their feedback within the process. The input of customers, particularly in the beginning of the process, is crucial for avoiding mistakes that are very difficult to fix later. The development process improvement suggested in the work is very relevant and there are no technical obstacles for it. The only obstacles are the deeply rooted habits in the companies.

The number of existing feature candidates at the moment is about 100. This consists of the unrealised feature candidates of old releases and some new feature candidates. It is only possible to implement approximately 10 % of them. By using empathic feature research, it would probably be possible to create solutions that would solve several problems at a time. In addition, by making use of operating task research and designing operating best practices, it would be possible to reduce the number of different operating task procedures. Furthermore, if the system itself were designed to support the best practices, many challenges would be met.

As a matter of fact, it is surprising that the suggested kind of empathic research is not used in this industry. According to the interviewee's experience, it is quite common in processing industry. There the supplier spends long periods of time with the operator, for example, in the commissioning phase. Obviously, the existing development process is the reason why the connection between network operating and network product development is not strong enough. The product development process should contain the empathic feature research, so that the *wanted* products could be developed instead of something that is imagined to be wanted.

The lack of redesign is one of the basic problems in this industry. First, the supplier sells something that does not exist, then they quickly create the first working version under strong deadline pressure so that the business can be started. When finally the functionality is up to the level that was sold to the customers, the redesign is omitted due to next deadline pressure and lack of resources. The system usability of this quickly designed functionality is not very good, particularly when there is no usability design at all in some products. As redesign is not done, the functionality is not improving even in the long run. The only case when a little redesign is visible, is when the technology generation changes – within the generation it is not done. Gradual redesign also within the generation would improve the system usability. However, the structure should be modular in order to assist the piece-by-piece redesign.

Operators want to operate the networks from process point of view. On the other hand, the network management systems have traditionally been designed from network element point of view (different worlds). For example, in the base station commissioning process, the operator has to do alterations into many systems, typically with many different tools. In addition, there are network elements and tools from many different vendors, as well as equipment from independent tool suppliers. There is also a suggestion in this work to design better operating task procedures, which could be recommended to operators. Designing the network and tools to support the recommended procedures would quite likely improve the situation. The interviewee has experience from another company, where they designed a network product to follow the customer process. This product became very popular and is currently used by more than 160 operators around the world. In his opinion, this is a good example of how designing the product to follow customer processes leads to a technical and commercial success.

It is a well-known fact that the supplier delivers the new BSS functionality without any immediate NMS support. This happens because the new network element software is usually completed before NMS software. The basic reason for this is that the NMS software testing can only be started after the BSS functionality exists. An easy solution would be not to release the BSS software before the NMS software is ready. In a practical competitive situation it is not possible to wait for completing the NMS software, since the new functionality must be quickly taken into use. Also the operators have competitors and they must start using the new functions early even without network management support. This leads to a situation, where the operators have to use rather low level management (e.g. MML) for handling the functionality. Later on when the NMS support is available, it is not necessarily used, since some way has been learned already. Ideal situation would be to deliver the whole functionality together, but so far the supplier has not succeeded in this.

There are many examples of too weak links between designers and users. Even the received operator feedback has not always reached the designers. Ideal would be, if the designers were experts in operating. As this is never the case in practice, the links should be stronger. The designers should spend some time in operating the supplier's own test network and also stay with the customers every now and then to follow the real operations. So far, the simultaneous designing and gathering of operating experience has not been successful due to the lack of resources. On the other hand, the products not turning out to be what they should is not good either. It should be possible somehow to arrange these stronger links. Perhaps the suggested way of using operator expertise in product development via the Internet would also be useful here.

Some misunderstandings in the validation

The planning effort in the definition of system usability was in one case understood to be on supplier side. This is not the case, although it is possible that the operator outsources the network planning as a service from the network supplier. Even in this case, it should be considered as operating effort, if the system usability is compared with a system where the operator does this planning. Of course, it can be left out, if it is reasonable in the studied case.

Synchronisation of subsystem releases was understood to mean synchronisation of different vendors' equipment. This is not the case – the synchronisation means one vendor's equipment. Likewise, the time pacing was misunderstood to mean different vendor's deliveries.

One person noted that: "Operators do not base their buying behaviour only on the best features, products or functionality. It is quite common that the buying decisions are based on political and commercial issues." Indeed this is the case, as it was described in the definitions. Since the reviewers did not see this part, the comment is understandable.

One comment was: "Effort cannot be directly calculated based on resources". This indicates that the effort was understood to be a calculated (estimated) figure, not measured. However, the idea is to measure the actual working hours. Still, it does not tell everything about the used effort, but this issue could be further refined in future research.

Reduction of user-observable complexity was understood to mean reduction of functionality. "Not necessar[il]y reduction, but hiding complexity. Reduction too often means less functionality." This is not what was meant by the concept. Reduction of user-observable complexity in this research does not imply that functionality should be reduced. Although, some of the interviewed operator people had nothing against reducing that either. Another reviewer commented that he did not understand the concept at all, since the referenced text (theoretical framework) was not available.

There was a comment concerning the use of multimedia in product development. "All the effort put to those could have been put in making the actual product". Based on the comment, it seems that the person had not understood the purpose of such descriptions in the product development context. As the main problem is how to create the *wanted* product and features, it is essential to receive guiding from the operators as early as possible, even before the product exists in any form. These multimedia descriptions and the operator feedback via the Internet is the means to achieve that.

Conclusion

Based on the comments of the validators, it seems that the results describe the real situation quite well. In the validation process, the theoretical (definitions) and empirical results were formulated into claims, which the validators commented. The comments revealed that they reviewed the presented claims primarily based on their own work experience. Those empirical results that were directly related to the every day work of the validators, invited majority of the comments. There was an overall agreement over the claims, except in some cases where something was misunderstood due to the concise information given to the reviewers. Theoretical results were commented less. The validators seemed to interpret them also through their work experience and were able to comment only some of them if somehow related to practical issues. The improvement suggestions were not commented very much – obviously because the validators could not relate to things that do not yet exist.

In the process, it also became clear how different the academic and industrial cultures really are. There was one person in the validation group, who understood the presented concepts and results better than others. Obviously, the reason for this is that the person was familiar with the theoretical framework and had been following the research by reading the researcher's writings once in a while. Based on the brief experiment, it would seem that getting familiar with the scientific research gradually by reading the researcher's work on progress as the research proceeds, the industrial managers are able to capitalise on the results in the work environment. Of course, this result is only trend setting, as the number of participants was very limited. If the objective is a fast exploitation of research results in developing engineering solutions, it might be good from the very beginning of the process to find key people for these liaison tasks in the organisation in question. These key people would follow the research as it proceeds and capitalise on the results after getting a good hold of the outcome. As this experiment shows, understanding complicated technical and theoretical scientific knowledge requires strong coevolutionary links between the different parties. If information is lost or changed in the links, the performance of the metasystem is not what it could be.

10.2 Evaluating the quality of the research

We use two approaches to evaluate the quality of the research: Reflective and objectivist. The reflective approach emphasises the interaction of the researcher, the researched phenomena and the research process (e.g. Lincoln/Guba 1985[197]). Thus, it is important to demonstrate how the researcher's interaction with the other two elements unfolds (e.g. Alvesson/Sköldberg 2000[4]). In this research, it is done by addressing three different aspects defined by Patton 1990[18]. The objectivist approach pursues to control the impact of the researcher on the results (Lincoln/Guba 1985[197]). More emphasis is placed on various control techniques to evaluate the research process (ibid.). This logic is addressed through the validity and reliability aspects proposed by Yin (Yin 1994[12]).

10.2.1 Reflective approach

1. What paradigm orientation and assumptions undergird the study?

Natural sciences are based on the assumption that external reality (outside of us) – *nature* if you will – exists regardless of us, and it can be indirectly observed by using different kinds of research methods (see for example Penrose 1989[196]). By using these methods, we try to learn the reality and build theoretical models of it in order to increase our understanding (refer to chapter 3.4 'Complex adaptive systems'). We must also remember that in hermeneutic research, the primary emphasis is on understanding, not on the object – subject arrangement. Knowledge is understood to cumulate as new research is based on previous findings, resulting in improved models little by little. In natural sciences, an improved model is considered to be the one that corresponds better with the results from empirical tests. In this research, the studied reality is the complex technical systems delivered to the operators, and the new understanding is created via studying the reality through a new theoretical framework.

The theoretical framework was developed during a long period of time and still evolved while doing the empirical research. It consists of many theories and aspects from biology and complexity research, but certainly can still be further developed. For example, the complex adaptive systems have been approached quite concisely. If the technology in the future adopts this kind of development, it would be useful to expand on this domain. At this point, when the theoretical framework is newly created and not been used in many studies, it is too early to say how well it fits the usability research. Further development requires more research.

System usability can be considered as a property of the system, although it can only be properly evaluated in connection with the users. Hence, we can state that by observing and interviewing the users (the indirect method), we can learn about the system usability (nature). The author considers system usability of complex technical systems as a more comprehensive issue than the present HCI-centric usability paradigm has asserted. Since the system usability concept includes e.g. the complexity aspect of the studied system, we could separately estimate the degree of complexity (number of interdependencies) in the system, and in this manner get a prediction of the system usability. How well this correlates with the true system usability is left for other studies to show.

2. What techniques and methods were used to ensure the integrity, validity and accuracy of the findings?

Two separate methods were used in the empirical data acquisition. Particularly, the observations by using the contextual inquiry method proved to be very revealing and gave a good view to the user's world. Lucas/Meech/Purcell 1997[166] also considered the contextual inquiry method excellent for user centred data gathering in their OMC usability research. The case descriptions were also reviewed and slightly corrected by two persons of the operating staff. The subsequent interviews complemented the observations. Also two different researchers were used to collect the empirical data. In addition, some flow charts from an operator describing some operating processes, as well as documentation of Hi-Tech Inc., such as NMS user's guides and documents concerning the product process, were used to enhance the view to this technology.

When planning the research, the intention was also to use a survey-type method by using a ready-made questionnaire for probing the usability of the NMS user interface. However, the users did not see it very meaningful, and many of them disposed of the forms without answering. Obviously, Hi-Tech Inc. had already used this method many times and perhaps the users were not very convinced of the usefulness of this kind of study. Anyway, this method would only have measured the user interface of the network management system (existing paradigm) and hence not produced much useful information for this research. As the users were not interested in the survey and the sample therefore became very small, it was decided to drop this method. Designing a new questionnaire for the purpose would have been a major overtaking and was not considered necessary, as the material acquired by the other methods was abundant and rich. This suggests that the methods were suitable for the research task.

It would have been good to test the systems of several technology suppliers in the research to be able to compare the level of system usability. However, this was not possible, as the operators refused to give information about other vendors' equipment. Anyway, it can be judged by the incidental comments that the system usability of the other suppliers' systems is not any better.

In the analysis phase, the first idea was to use the operators as the cases, but later on it was decided that the operating tasks are more suitable for the purpose. According to Stake 1995[16], a case is a specific, a complex, functioning thing. Although, both case possibilities fulfil these requirements, the operator tasks are more concrete and descriptive as cases, if we are studying the system usability of the technology used for these tasks. As it has been pointed out, usability must be studied in connection with the real context, hence also the operating tasks are a more natural choice for cases. We could say that the tasks are closer to the problems of system usability than the more vague concept of operator. Operating tasks being smaller "bounded systems" (Stake 1995[16]) also allows a higher resolution to study the problem. For the mentioned reasons, it was decided to use the tasks as the cases in spite of the fact that Stake considers events and processes fitting the definition less well.

Validity is also improved in the evaluation made after the research, where a group of managers from Hi-Tech Inc., an operator manager and a complex system researcher reviewed the main results and improvement suggestions. Although this kind of validation is not usually done in hermeneutic research, we had certain research interests in mind when doing it. In hermeneutic research it is more important to discuss arguments and counterarguments – *logic of argumentation* instead of *logic of validation* (Alvesson/Sköldberg 2000[4]).

3. What does the researcher bring to the study in terms of qualifications, experience and perspective?

The researcher has an M.Sc.-degree in telecommunications technology and a long experience (more than a decade) with cellular networks, seven years of which was obtained in radio network planning of these systems. As network planning is done in close co-operation with the operators in the country in question and many times in their premises, it gives a good view to the world of network operating. This kind of intimate relationship cannot be substituted for e.g. by working in the office of the supplier and occasionally interviewing the operators.

The researcher took a familiar technical point of view to the research and not e.g. a psychological or sociological approach that would have been more alien considering the technical background. Consequently, the problem was also seen, above all, as a technological dilemma. The researcher is also employed by an industrial company manufacturing complex technical systems, and hence the technological focus was a natural choice. The system supplier creates the system usability (good or bad) when building the technical system, so good understanding of technology, also from novel viewpoints, is necessary. New perspective to the problem was gained by using a rather unconventional theoretical framework particularly created for this research. This perspective gives a new view to usability of complex technical systems, as it builds on the theories of how nature creates complex systems. Those systems are, unquestionably, usable systems.

10.2.2 Objectivist approach

Yin (1994[12]) presents four tests to judge the quality of empirical case study research: Construct validity, internal validity, external validity and reliability (see Table 16).

Table 16. Research design tests

Tests	Case study tactic	Phase of research
construct validity	use multiple sources of evidence establish chain of evidence have key informants review draft case study report	data collection data collection composition
internal validity	do pattern matching do explanation building do time series analysis	data analysis data analysis data analysis
external validity	use replication logic in multiple case studies use case study protocol	research design data collection
reliability	develop case study database	data collection

Construct validity

Construct validity refers to establishing correct operational measures for the concepts being studied. From the three measures presented in the table, the first one using multiple sources of evidence was already discussed under question no. 2 by Patton above.

Establish chain of evidence. Yin sees the chain of evidence as moving from one portion of the study to another with clear cross-referencing to methodological procedures and to the resulting evidence. In this study, due to the adopted hermeneutical approach aiming at understanding the researched phenomena, the chain of evidence was established between the research questions, theoretical framework, empirical study, as well as results. There are two different sets of research questions in the study. The first two research questions were addressed by exploring research literature of complexity research and natural complex systems, as well as by creating the theoretical framework based on the exploration. The framework was then used to introduce the concept of system usability to reconceptualise usability study of complex technical systems to better account for the complex characteristics of those systems. The third research question with three sub questions addresses the empirical issues of system usability. First, the complexity of cellular networks was established by investigating interdependencies of various parameters in a simulated environment. Finally, an empirical study was conducted to apply the theoretical framework for studying system usability of the cellular networks. The role of the empirical research was two-fold, a) on the one hand, it validated the explorative theoretical framework and b) on the other hand, explained the empirical phenomena in a novel way. The research data was analysed by using concepts from the theoretical framework thereby linking the results of the empirical analysis to the theoretical reconceptualisation of usability.

Have key informants review draft case study report. The likelihood of falsely reporting observed phenomena was reduced by giving the case descriptions for inspection to the operator, where two persons checked them and made a few minor modifications. As revealed in the study, the operators and suppliers have different standpoints on network operating. When different renditions exist, the procedure helps to identify the various perspectives. Network professionals from cellular network industry also read the results and their feedback was used for complementing the picture in the validation phase.

Internal validity

According to Yin, internal validity refers to establishing a causal relationship, whereby certain conditions are shown to lead to other conditions. However, in hermeneutical research, relations between observations are established through interpretation, which aims at understanding the pattern of relationships in the data (see Lincoln/Guba 1985[197]). In this research, we did not search for causalities in the data, instead we interpreted the observations by using the theoretical framework and thus gained understanding of the phenomena. By proceeding with higher level interpretations, we also improved the external validity.

Pattern matching was used on high level to find evidence for a) the system usability of cellular networks is not very good, b) the systems are quite complex, as well as distributed and require a more comprehensive usability approach than has been used. The alternative was that a) the system usability of the systems is fine and hence b) the conventional HCI-oriented usability is sufficient for these complex technical systems. *Explanation-building* was not used, as the research was not an explanatory study. *Time series analysis* is not relevant for this research, as it is not a longitudinal study.

External validity

According to Yin, external validity establishes the domain to which the findings of the study can be generalised. In qualitative research however, we do not seek an explanation for the empirical sample (non-representative sample in qualitative research) from some imaginary basic set to which the results should apply. Instead, according to Alasuutari (1994[198]), in qualitative research the prime objective is local explaining. The explanation must apply to the empirical material, which it is based on. The explanation is searched for the essential features (from research relevance point of view) of a phenomenon constructed with the aid of empirical examples.

We used eight different exemplary operating task cases and four different operators in the research. Based on exemplary results, it is also possible to create a typology, which is completed by a logical derivation to a complete description of all conceivable phenomena within the context – even variant ones and those that were not present in the empirical data. However, we should not use the term generalisation in connection with qualitative research, since this word is reserved for survey research. Rather, commensuration would be a better term. When we concentrate on a certain limited phenomenon in local interpretation, commensurating this phenomenon to a larger whole is a kind of extrapolation, where we conceptually take possession of a larger field of phenomena (ibid.).

Showing the metalevel relevance is another task. In this work, we have taken the interpretations to a higher level by merging them into a few meta-conclusions. As many occasions of the themes have been collected and the results have been interpreted on metalevel, we are able to commensurate the results to cellular network domain. There is also reason to believe that the found phenomena are not unique and proprietary to cellular networks, but also apply to other complex technical systems at least to some extent. This is evidenced by the comments of the researcher having experience on other complex technical systems. According to her, many of the described phenomena have also been cited in connection with other complex systems.

At this point, we do not have experience on how well the created theoretical framework can be applied to all complex technical system research, but it seems reasonable that a framework using a complexity understanding approach would apply to various complex technical systems – after all, complexity is a universal natural phenomenon. It could probably also be quite generally applied to usability research – even user interfaces have user-observable complexity. However, very simple devices would probably not benefit from this approach.

Reliability

Many times, the research methodology literature mentions replicability as the main feature in reliability. However, replicability reflects more the assumptions of quantitative inquiry. Marshall/Rossman (1989[199]) state that qualitative research does not pretend to be replicable. Although the replicability is not an issue, systematic inquiry must contain means to make data available in case the findings are challenged or researcher wants to reanalyse the material (ibid.). In this research, also the validation made after the research corroborated the results.

Use case study protocol. Yin suggests a written protocol, which contains the procedures and general rules that should be followed in the study. The author did not compose such a document. However, a research plan was drawn, where the purpose of the study was stated and methodological issues concerning the project were considered. Relevant theoretical readings were explored and a preliminary theoretical framework was developed before the fieldwork. The themes guiding the data collection were based on the framework and the author's work experience. However, the interviewees had a relatively large influence on the contents and importance of the topics in the discussions. The fieldwork and report structure issues are described in the methodology section in the beginning of the research report.

Develop case study database. Yin identifies four components in developing a database: notes, documents, tabular material and narratives. The first two are relevant for the study. In regard to notes, all the interviews and observations were recorded and transcribed for later analysis. They are stored together with the original audio files, permissions and background descriptions of the informants. Filed documents consist of some flow charts of operating procedures from operator A and basic operating principles and procedures of Hi-Tech Inc.

11. DISCUSSION AND FURTHER RESEARCH

The classic engineering approach is considering only functionality. To solve a problem, engineers create technical devices to offer the required functionality, and traditionally usability has not been a highly regarded engineering consideration. In cellular networks, the usability approach has been quite narrow and these aspects have mainly been considered in user interface design only. This is not capable of mending all the neglects in usability elsewhere in the system. As the usability has been understood mainly via user interfaces, these considerations have been left to the developers of the network management system. In the system itself everything is designed just to meet the functional specifications, and since the usability aspects of the BSS have not been specified, usability design is omitted. With the words of Nigel Bevan (Bevan 1995b[200]): "In software engineering, the conventional objective for quality is to build a software product, which meets the specification. However, this alone is rarely sufficient to ensure quality of use – that the product can be used for its intended purpose in the real world".

The user-observable complexity of the systems is increasing all the time, if new features are introduced without concern of the complexity issues or system usability. If we consider the complexity increase over the cellular technology generations, it is apparent that the forthcoming technology will be much more complex than the present. If this development continues, it also means that the user effort increases and the networks will be even further away from the optimum, since the ever-increasing user-observable complexity makes the systems more difficult for the users to handle.

Complexity is a natural phenomenon, the implications of which are visible in all complex systems (see e.g. Kauffman 1992[126]). In nature, the systems are created spontaneously, but they obey the laws of complexity automatically. It would be useful to apply our understanding of the natural complex systems to technology when building complex technical systems, since understanding complexity gives possibilities to manage it and avoid catastrophes. In these systems, it is not sufficient to handle system usability by applying the HCI-approach only. The inherent complexity of the system should also be considered from the very beginning, perhaps starting already from the standardisation phase.

Standardisation was not in the scope of this research, but it certainly has an impact on system complexity. Traditionally in telecommunication systems, the standardisation has been using a top-down approach (3GPP/ETSI). This means that the standardisation has tried to secure consistency of all views of the system and its components. Generally, competing solutions for the same problem have not been approved. The Internet world, on the other hand, has followed a bottom-up approach

(IETF). This means that problems are solved one-by-one, nobody is keeping track of the complete system fitness and operators do more system integration. From these two approaches, the latter one is closer to the natural (co)evolution and the framework of this research. However, in nature, there is no option in this sense, because there is nobody to plan and supervise the overall development. Although nature produces good results in the long run with the bottom-up approach, reaching these solutions can waste a lot of material (experimental phenotypes) in the process. In man-made technology, this is acceptable in SW development, as it does not waste any tangible material.

After standardisation, the technology supplier's R&D organisation faces the complexity challenge. In cellular network development, it is typically the BSS unit, which drives the design and NMS development follows behind. This marching order comes from the assumption that NMS designers must first know the BSS features to be managed. The other possibility is that the NMS design would drive the business in the name of improving the network operating. On the other hand, many solutions should be market tested first, as it is reasonable to optimise the NMS for the successful solutions only. Probably though, the BSS and NMS developments are thought excessively as distinct issues. System usability is not only an NMS issue, as it might seem today. System usability has a lot to do with BSS (or other parts of the network). Considering smaller pieces of functionality from both aspects at a time and building them together could be a better way of handling the system development. By doing this, it would also be possible to consider implementing as much autonomous functionality into the BSS elements as possible.

In this research, the approach has been to use the operator view. In this point of view prioritising the individual NMS features is not as relevant as it is in the (present) supplier point of view. It would seem that the operators normally prefer applications that enable accomplishing the tasks (e.g. creating or commissioning the BTSs or rehosting the BTSs under another BSC etc.) with low manual effort and without intervention of many work groups. This would require task oriented applications and rather high level approach (metasystem transition) with high level of automation. Although prioritising the NMS features has not been studied, it seems that the operators would like to think the functionalities as comprehensive packages. In other words, part of it can be in BSS, another part in NMS, but they are not interested in prioritising only a part of the functionality (e.g. NMS feature supporting some BSS function). But as mentioned, there is no conclusive evidence for this, and hence it is not discussed in the text.

Practically all operators function in a multivendor environment. This is not very visible in this research, since it was not possible to study several vendors' systems due to access reasons. Most operators have more than one BSS supplier, typically a certain contiguous geographical area is covered by one supplier's equipment. The suggested harmonisation of operating task procedures means harmonising the procedures for all operators within such an area. Different vendors have different equipment, and hence cannot be handled with the same kind of procedures. Suppliers must often assume that their NMS is not the only solution the operators are using. In the current thinking, this means that suppliers should keep their NMS generic and offering only basic functionality. This, in turn enforces the thinking among operators that other equipment are required to complement the system. With one particular vendor's equipment, the vendor could find one or a few ways, which are the most efficient. If the recommended ways of performing the tasks are more efficient than the used ones, it would not be wise to reject them. This issue is connected with the system development. It does not pay to only recommend new practices, unless the system is built so that is fully supports the defined task practices. In the mentioned case, the synergy of operating tasks and the system properties could bring significant improvements in system usability.

Keinonen 1998[93] argues that usability of simple devices can be understood as a one-dimensional concept, where devices are either usable with few functions or versatile with many functions. This is probably true at the moment even for cellular telephone networks. This can be due to the present usability paradigm, which does not address the structural complexity of the systems. Understanding usability with the aid of complexity allows for good usability even in complex and plenty of functionality containing technical systems. The solution space in complex technical systems is so huge that there are always excellent solutions also for usability, albeit not very easily found. Using the system usability approach means that main focus be directed in designing the system itself based on certain sound principles and doing the finishing touch with the user interfaces. However, this does not mean that user interfaces should be designed last after everything else. User interface designers live in coevolution with the other designers from the beginning, but now the system design has a new ingredient: complexity understanding. If we do not understand the significance of complexity and manage it, we cannot build very usable complex technical systems.

Summary

In this research, we have done the following.

a) Shown that the present system usability of cellular networks is not very good
b) Created a framework for understanding system usability of complex technical systems in a new way
c) Applied the coevolution theory for understanding product development of complex technical systems as a coevolutionary process
d) Created a definition and a measuring idea for system usability
e) Created a set of guidelines to improve system usability

In this research, we have charted the present system usability situation of cellular networks. The results show that the system usability of these systems is presently not very good, as the user-observable complexity is high and the operational work must be done by investing a lot of low level manual effort in doing the tasks. Some main reasons for this seem to be incomplete understanding of operational work in the suppliers' engineering organisation and the present usability paradigm, which understands usability mainly via the HCI issues.

The developed theoretical framework provides a new way of approaching usability. As shown in the research, the present paradigm of usability has not been able to create very usable complex technical systems. The created approach provides new potential in improving this situation. The framework also improves the understanding of turbulent hi-tech product development by studying technical evolution with the aid of the dynamic adaptive landscape, where error and complexity catastrophes are lurking the companies.

Based on the combined understanding gained from the theoretical framework and the empirical insight, we have applied the coevolution theory to provide a model for understanding technical development as a coevolutionary process. This coevolution occurs at many different levels, for example, company level involving customers and competitors and product level involving the subunits and components in a certain product.

We have also provided new contribution for usability theories to better grasp highly specialised complex technical systems. The model defines system usability with the aid of the effort the different user groups have to invest into the daily operations with the system. Improvements in system usability can be understood either as a system usability *maximising* task in the usability landscape or an effort *minimising* task in the effort landscape.

Last, but not least, we have defined a set of guidelines to improve the system usability of complex technical systems. By following these rules, the companies producing complex technical systems can potentially improve the system usability of their products, which would show in reduced operating effort (see Figure 43).

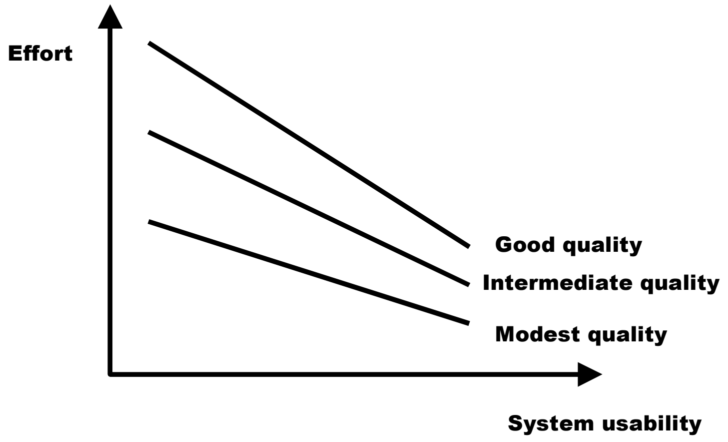


Figure 43. A graph characterising operator effort as a function of system usability.

Further research

In this research, we have introduced the concept of system usability to point out that the core issues in improving the system usability of complex technical systems are a) understanding the system complexity, b) making sure that the complexity will not hinder attaining good system usability. The focus of the study has been on technical aspects of complexity. However, as Vicente 1999[116] points out, there is also the social aspect of complexity. Particularly, in coevolutionary development of technology, where several different companies are involved, the organisational complexity is a significant success factor. It would be useful to study the complexity of the coevolutionary product development from human organisation point of view. For example, what kind of N and K-values (also C and S) exist in the practical organisations of the suppliers and operators. This would give a more complete picture of complexity in technical product development by creating an insight to the interdependence of different development teams, as well as manufacturing and marketing.

The present usability literature understands usability mainly through user interfaces. The author has created a theoretical framework for the research based on biological theories of natural complex systems, complexity research, usability and technological insight. As the theoretical framework is newly created, it would be useful to evaluate its suitability for usability research. This could be studied as a comparative research. After a usability assessment based on the conventional usability, another assessment would be made to the same system and the results compared. This could be taken even further by making two new versions of the system based on the improvement suggestions and then comparing the systems.

In this work, we only studied the system usability of one network supplier's equipment due to access problems, as the operators were not willing to give information concerning all vendors.

However, it would be very useful to know how each vendor's equipment compare with each other. The developed system usability metrics could be used in this research. Knowing how the system usability of each vendor's equipment relates to the others, could, for example, improve the competition between vendors and in this way enhance the system usability of all vendors' equipment. Strong coevolution benefits all players.

In this work, we defined the system usability with the aid of the effort, which the users must invest in operating the system. This definition could also be used for measuring the system usability of various complex technical systems. As we did not perform any such measurements in this work, it would be useful to learn more about measuring and comparing different systems. This would include comparing the networks of different cellular network operators and also other complex technical systems, as well as further developing the metrics. The study could also assess the N and K-values between the different efforts and also the N and K-values between different system features to see how they affect system usability. Knowing things like that would enable forming an estimate of system usability even without measuring.

In this work, we created a set of guidelines for improving the system usability of complex technical systems. As this work did not contain applying the created guidelines to practice, it would be good to test them in practice in the product development of cellular networks and other complex technical systems. Testing the guidelines leads to a change process, where new organisational practices and ways of working are developed. The process could be realised as action research. (e.g. Kemmis/McTaggart 2000[201]). Action research is particularly suitable for the task, since it allows system usability researchers and product developers to share their expertise and to simultaneously learn and implement the acquired new knowledge.

References (There is also another list ordered by the reference indexes)

ACM SIGCHI (1992), Curricula for Human-Computer Interaction. At http://www.acm.org/sigchi/cdg (9.4.2002)

Adams, E. & Willets, K. (1996), The lean communications provider. Surviving the shakeout through service management excellence. McGraw-Hill, New York.

Alasuutari, P. (1994), Laadullinen tutkimus. Vastapaino, Tampere.

Albers, J. (1987), Interaction of color. Yale University Press, New Haven.

Alvesson, M. & Sköldberg, K (2000), Reflexive methodology. New vistas for qualitative research. Sage Publications, Thousand Oaks.

Anderson, P. & Tushman, M. (1990), Technological discontinuities and dominant design: A cyclical model of technological change. Administrative Science Quarterly 35, pp. 604-633.

Apple Computer (1992), Macintosh human interface guidelines. Addison-Wesley, Reading.

Arlov, L. (1997), GUI Design for dummies. IDG Books, Foster city.

Arthur, W. (1993), Why do things become more complex? Scientific American, May 1993, p. 144.

Ashby, W. (1958), Requisite variety and implications for control of complex systems. Cybernetica 1, pp. 83-99.

Baecker, R. & Grudin, J. & Buxton, W. & Greenberg, S. (editors) (1995, 2nd edition), Readings in Human-Computer Interaction: Toward the year 2000. Morgan Kaufmann Publishers, San Francisco.

Bain, S. & Gray, D. (1996), Looking good online: The ultimate resource to creating effective web designs. Ventana, Research Triangle Park.

Baldwin, C. & Clark, K. (2000), Design rules: The power of modularity. MIT Press, Cambridge.

Baldwin, C. & Clark, K. (1997), Managing in an age of modularity. Harvard Business Review, vol. 75, no. 5, pp. 84-93.

Bartlett, F. (1961), Remembering. Cambridge University Press, Cambridge.

Baum, J. & Singh, J. (1994), Organization-environment coevolution. In J. Singh (editor): Organizational evolution: New directions. pp. 379-401. Sage Publications, Newbury Park.

Beinhocker, E. (1997), Strategy at the edge of chaos. The McKinsey Quarterly, no. 1, 1997, pp. 24-39.

Bendett, R. & Neelakanta P. (2000), A relative complexity metric for decision-theoretic applications in complex systems. Complex Systems vol. 12, no. 3, pp. 281-295.

Bertalanffy, L. von (1968, revised edition 1984), General system theory. George Braziller Inc, New York.

Bevan, N. (1995a), Measuring usability as quality of use. Journal of Software Quality, Issue 4, pp. 115-140.

Bevan, N. (1995b), Usability is Quality of Use. In Y. Anzai and K. Ogawa (editors): Proceedings of the 6th international conference on human computer interaction, Yokohama, July 1995. Elsevier, Amsterdam.

Beyer, H. & Holtzblatt, K. (1998), Contextual design: Defining customer-centered systems. Morgan Kaufmann, San Francisco.

Bias, R. & Mayhew, D. (editors) (1994), Cost-justifying usability. Academic Press, Boston.

Bickford, P. (1997), The art of developing easy-to-use software. AP professional, Boston.

Blomberg, J. & Giacomi, J. & Mosher, A. & Swenton-Hall, P. (1993), Ethnographic field methods and their relation to design. In D. Schuler and A. Namioka (editors): Participatory design: Principles and practices, pp. 123-155. Lawrence Erlbaum Associates, Hillsdale.

Bohm, D. (1980), Wholeness and the implicate order. Routledge, London.

Bonner, J. (1988), The Evolution of complexity. Princeton University Press, Princeton.

Borenstein, N. (1994), Programming as if people mattered: Friendly programs, Software engineering and other noble delutions. Princeton University Press, Princeton.

Britannica.com Inc (2001), Encyclopedia Britannica. At http://www.britannica.com (9.4.2002)

Brodie, E. (1997a), Evolution course. Course material. University of Indiana. Department of Biology. At http://www.bio.indiana.edu/courses/L318-brodie/L318/L318lect16-fitness.html (9.4.2002)

Brodie, E. (1997b), Evolution course. Course material. University of Indiana. Department of Biology. At http://www.bio.indiana.edu/courses/L318-brodie/L318/L318lect27coevolution.html (9.4.2002)

Brown, S. & Eisenhardt, K. (1998), Competing on the edge – Strategy as structured chaos. Harvard Business School Press, Boston.

Browne, D. (1994), STUDIO: Structured user-interface design for interaction optimisation. Prentice Hall, New York.

Buur, J. & Bodker, S. (2000), From usability lab to design collaboratorium: Reframing usability practice. Designing Interactive Systems Conference (DIS 2000), ACM Press, New York.

Card, S. & Moran, T. & Newell, A. (1983), The psychology of human-computer interaction. Lawrence Erlbaum Associates, Hillsdale.

Carey, J. (editor) (1997), Human factors in information systems: The relationship between user interface design and human performance. Ablex, Greenwich.

Casey, S. (1998, 2nd edition), Set phasers on stun – And other true tales of design, technology and human error. Aegean Publishing Company, Santa Barbara.

Chapanis, A. (1996), Human factors in system engineering. Wiley, New York.

Christensen, W. & Collier, J. & Hooker, C. (2000), Adaptiveness and adaptation: A new autonomy-theoretic analysis. The University of Newcastle – Australia.
At http://www.newcastle.edu.au/centre/casrg/members/christensenwayne/Ch3-Adaptiveness.pdf (9.4.2002)

Christensen, W. & Hooker, C. (1999), Autonomy and the emergence of intelligence: Organised interactive construction. The University of Newcastle – Australia. At http://www.newcastle.edu.au/centre/casrg/publications/AUTO-NPO.pdf (9.4.2002)

Cliffe, S. (1998), Knowledge management: The well-connected business. Harvard Business Review (Briefings from the editors). July-August 1998.

Coe, M. (1996), Human factors for technical communicators. Wiley, New York.

Colby, C. (1996), Introduction to evolutionary biology. At http://www.talkorigins.org/faqs/faq-intro-to-biology.html (9.4.2002)

Collins, D. (1995), Designing object-oriented user interfaces. Benjamin/Cummings Publishing, Redwood City.

Cooper, A. (1995), About face: The essentials of user interface design. IDG Books Worldwide, Foster City.

Cooper, R. (1990), Stage-Gate Systems: A New Tool for Managing New Products. Business Horizons May-June 1990, pp. 44 -54.

Course material. Brown University. Division of Biology and Medicine. Department of Ecology and Evolutionary Biology. At http://biomed.brown.edu/Courses/BIO48/27.Coevolution.HTML (9.4.2002)

Crabtree, A. & Nichols, D. & O'Brien, J. & Rouncefield, M. & Twidale, M. (2000), Ethnomethodologically informed ethnography and information system design. Journal of the American society for information science. May 2000 pp. 666-682.

Craik, K. (1943), The nature of explanation. Cambridge University Press, Cambridge.

Crutchfield, J. & Nimwegen, E. (1999), The evolutionary unfolding of complexity. Santa Fe Institute working paper 99-02-015. At http://www.santafe.edu/sfi/publications/wpabstract/199902015 (9.4.2002)

Cusumano, M. & Yoffie, D. (1998), Competing on Internet time: Lessons from Netscape and its battle with Microsoft. Free Press/Simon & Schuster, New York.

Dahan, E. & Hauser, R. (2000), The virtual customer: Communication, conceptualization, and computation. MIT Sloan School of Management. At http://mitsloan.mit.edu/vc/Pages/papers.html (9.4.2002)

De Vos, G. (1997), The Alcatel SDH network release concept. Alcatel Telecommunications Review, 1st Quarter 1997, pp. 19-23.

Dix, A. & Finlay, J. & Abowd, G. & Beale, R. (1998, 2nd edition), Human-Computer Interaction. Prentice Hall Europe, London.

Dragut, A. & Bertrand, J. (2001), Time and resource characteristics of radical new product development (NPD) projects and their dynamic control. In Proceedings of the 12th annual conference of the production and operations management society, March 30-April 2, 2001, Orlando.

Dumas, J. & Redish, J. (1999, revised edition), A practical guide to usability testing. Intellect Ltd, Bristol.

Eberts, R. (1995), User interface design. Prentice Hall, Englewood Cliffs.

Edmonds, B. (1995), What is complexity? – The philosophy of complexity *per se* with application to some examples in evolution. In F. Heylighen and D. Aerts (editors): The Evolution of Complexity, Kluwer Academic, Dordrecht. At http://www.cpm.mmu.ac.uk/~bruce/evolcomp/evolcomp_1.html (9.4.2002)

Ehrlich, P. & Raven, P. (1964), Butterflies and plants: A study in coevolution. Evolution 18, pp. 586-608.

Eigen, M. & Schuster, P. (1979), The Hypercycle - A principle of natural self-organisation. Springer-Verlag, Berlin.

Eisenhardt, K. (1999), Strategy as strategic decision making. Sloan Management Review, vol. 40, no. 3.

Encarta Reference. Microsoft encyclopedia at http://encarta.msn.com/reference (9.4.2002)

Eskikaya, B. (1998), A glimpse at artificial coevolution. Course material. University of Sussex at Brighton. At http://www.cogs.susx.ac.uk/lab/nlp/gazdar/teach/atc/1998/talksnode25.html (9.4.2002)

ETSI specification ETR095 (1993), Human factors (HF); Guide for usability evaluations of telecommunications systems and services. European Telecommunications Standards Institute. At http://www.etsi.org/ (9.4.2002)

ETSI specification ETS 300 615 (1996), Digital cellular telecommunications system (Phase 2) (GSM): Performance data measurements (GSM 12.04), European Telecommunications Standards Institute. At http://www.etsi.org/ (9.4.2002)

ETSI specification EG 201 472 (2000), Human factors (HF): Usability evaluation for the design of telecommunication systems, services and terminals. European Telecommunications Standards Institute. At http://www.etsi.org/ (9.4.2002)

Fleming, L. & Sorenson, O. (1999), Technology as a complex adaptive system: Evidence from patent data. Forthcoming in Research Policy. At http://www.anderson.ucla.edu/faculty/olav.sorenson/fs99.pdf (9.4.2002)

Fleming, L. & Sorenson. O. (2001), Science, Technological interdependence, and invention. Harvard Business School Working paper.

Fosythe, C. & Grose, E. & Ratner, J. (editors) (1998), Human factors and web development paperback. Lawrence Erlbaum, Mahwah.

Fowler, S. (1998), GUI design handbook. McGraw-Hill, New York.

Funes, P. & Sklar, E. & Juillé, H. & Pollack, J. (1998), Animal-animat coevolution: Using the animal population as fitness function. In R. Pfeifer et. al. (editors): From animals to animats 5: Proceedings of the fifth international conference on simulation of adaptive behavior. MIT Press. Pp. 525-533. At http://www.demo.cs.brandeis.edu/papers/ (9.4.2002)

Gardiner, M. & Christie, B. (editors) (1987), Applying cognitive psychology to user-interface design. Wiley & Sons, Chichester.

Geertz, C. (1979) From the native's point of view: On the nature of anthropological understanding. In P. Rabinow and W. Sullivan (editors), Interpretive social science: A reader, pp. 225-241. University of California Press, Berkeley.

Gell-Mann, M. (1994), The Quark and the Jaguar – Adventures in the simple and the complex. Little, Brown and Company, London.

Gell-Mann, M. (1995), What is complexity? Complexity vol. 1, no. 1, pp. 16-19.

Gell-Mann, M. & Lloyd, S. (1996), Information measures, effective complexity, and total information. Complexity, vol. 2, no. 1, pp. 44-52.

Gell-Mann, M. (2002), Plectics: The study of simplicity and complexity. Europhysics News, vol. 33, no. 1, pp. 17-20.

Gitzendanner, M. (2000), Principles of organic evolution. Course material. Washington State University. Ecology and Evolutionary Biology. At http://www.wsu.edu/~mgitzend/evolution/Notes15b/Notes15b.pdf (9.4.2002)

Greenbaum, J. & Kyng, M. (editors) (1991), Design at work: Cooperative design of computer systems. Lawrence Erlbaum, Hillsdale.

Gubrium, J. & Holstein, J. (2002), From the individual interview to the interview society. In J. Gubrium and J. Holstein (editors): Handbook of interview research: Context and method. Sage Publications, Thousand Oaks.

Hamel, G. (2001), Revolution vs. evolution: You need both. Harvard Business Review, vol. 79, no. 5, pp. 150-153.

Hansen, M. (1999), The search-transfer problem: The role of weak ties in sharing knowledge across organisation subunits. Administrative Science Quarterly, March 1999.

Helander, M. & Landauer, T. & Prabhu, P. (editors) (1997), Handbook of Human-Computer Interaction, second edition. North-Holland, Amsterdam.

Heylighen, F. (1994), Fitness as a default: The evolutionary basis for cognitive complexity reduction. In R. Trappl (editor): Cybernetics and Systems, World Science, Singapore, pp.1595-1602.

Heylighen, F. (1999), The growth of structural and functional complexity during evolution. In F. Heylighen, J. Bollen and A. Riegler (editors): The evolution of complexity, Kluwer Academic, Dordrecht, pp. 17-44. At http://pespmc1.vub.ac.be/papers/ComplexityGrowth.html (9.4.2002)

Hi-Tech Inc. FM (1998), NMS For Managing Cellular Networks: Fault Management Basic Operating Principles and Procedures. Not public.

Hi-Tech Inc. CM (1998), NMS For Managing Cellular Networks: Configuration Management Basic Operating Principles and Procedures. Not public.

Hi-Tech Inc. PM (1998), NMS For Managing Cellular Networks: Performance Management Basic Operating Principles and Procedures. Not public.

Holland, J. (1995), Hidden order – How adaptation builds complexity. Addison-Wesley, Reading.

Hom, J. (1996), Usability methods toolbox. At http://www.best.com/~jthom/usability/ (9.4.2002)

Horton, W. (1992), Illustrating computer documentation: The art of presenting information graphically on paper and online. Wiley, New York.

Horton, W. (1994), The icon book: Visual symbols for computer systems and documentation. Wiley, New York.

Howlett, V. (1996), Visual interface design for Windows. Wiley, New York.

Iansiti, M. & MacCormack A. (1997), Developing product on internet time. Harvard Business Review, vol. 75, no. 5, pp. 108-117.

ISO 9241-11 (1998), Ergonomic requirements for office work with visual display terminals (VDTs). Part 11: Guidance on usability. At http://www.iso.ch/iso/en/ISOOnline.frontpage (9.4.2002)

ISO/IEC 9126-1 (2001), Software engineering - Product quality. Part 1: Quality model. At http://www.iso.ch/iso/en/ISOOnline.frontpage (9.4.2002)

Jeffress, L. (editor), (1951), Cerebral mechanisms in behaviour - The Hixon symposium. Wiley & Sons, New York.

Jirotka, M. & Goguen, P. (editors) (1994), Requirements engineering --- Social and technical issues. Academic Press, London.

Johnson, J. (2002), In-depth interviewing. In J. Gubrium and J. Holstein (editors): Handbook of interview research: Context and method. Sage Publications, Thousand Oaks.

Kadoda, G. (2000), A cognitive dimensions view of the differences between designers and users of theorem proving assistants. In A. Blackwell and E. Bilotta (editors): Proceedings of 12th workshop of the psychology of programming interest group, Cozenza, April 2000, pp. 33-44.

Kaneko, K. & Tsuda, I. (1994), Constructive complexity and artificial reality: An introduction. Physica, D 75, pp. 1-10.

Karat, J. (editor) (1991), Taking software design seriously: Practical techniques for Human-Computer Interaction design. Academic Press, Boston.

Kauffman, S. (1992), The origins of order – Self-organization and selection in evolution. Oxford University Press, New York.

Kauffman, S. & Lobo, J. & Macready, W. (1998), Optimal search on a technology landscape. Santa Fe Institute working paper 98-10-091E. At http://www.santafe.edu/sfi/publications/wpabstract/199810091 (9.4.2002)

Kauffman, S. & Macready, W. (1995), Technological evolution and adaptive organizations. Complexity, vol. 1, no. 2, pp. 26-43.

Kauffman, S. (1995), At home in the universe – The search for laws of complexity. Penguin books, London.

Kauffman, S. (2000), Investigations. Oxford University Press, New York.

Keinonen, T. (1998), One-dimensional usability. University of Art and Design Helsinki, UIAH Publications, Helsinki.

Kemmis, S. & McTaggart, R. (2000), Participatory action research. In N. Denzin and Y. Lincoln (editors): Handbook of qualitative research, 2nd edition, pp. 567-605. Sage Publications, Thousand Oaks.

Koestler, A. (1964), The act of creation. Penguin Books, London.

Kyng, M. & Mathiassen, L.(editors) (1997), Computers and design in context. MIT Press, Cambridge.

Landauer, T. (1995), The trouble with computers: Usefulness, usability, and productivity. MIT Press, Cambridge.

Lansdale, M. & Ormerod, T. (1994), Understanding interfaces: A handbook of human-computer dialogue. Academic Press, London.

Leonard, D. & Rayport, J. (1997), Spark innovation through empathic design. Harvard Business Review, November-December 1997.

Lester, R. & Piore, M. & Malek, K. (1998), Interpretive management: What general managers can learn from designers. Harvard Business Review, March-April 1998.

Lincoln, Y. & Guba, E. (1985), Naturalistic inquiry. Sage Publications. Newbury Park.

Lindgaard, G. (1994), Usability testing and system evaluation: A guide for designing useful computer systems. Chapman & Hall, London.

Lucas, M. & Meech, F. & Purcell, C. (1997), Re-engineering a complex network interface. In M. Smith, G. Salvendy and R. Koubek (editors): Design of computing systems: Social and ergonomic considerations. Proceedings of the 7th international conference in Human-Computer Interaction, vol. 2. San Francisco, USA. Elsevier, Amsterdam.

MacCormack, A. (2000), Towards Contingent Model of the New Product Development Process: A Comparative Empirical Study. A working paper. Harvard Business School, Boston.

Madsen, K. & Borgholm, T. (1999), Cooperative Usability Practices. Communications of the ACM vol. 42, no. 5.

Maguire, S. & McKelvey, W. (1999), Complexity and management: Moving from fad to firm foundations. Emergence, vol. 1, no. 2.

Mandel, T. (1994), The GUI-OOUI war: Windows vs OS/2. Van Nostrand Reinhold, New York.

Marcus, A. & Smilonich, N. & Thompson, L. (1995), The cross-GUI handbook. Addison-Wesley, Reading.

Marcus, A. (1991), Graphic design for electronic documents and user interfaces. Addison Wesley, New York.

Markoff, J. (2001), Software's next leap is out of the box. New York Times, June 13, 2001. At http://www.nytimes.com/2001/06/13/technology/13MARK.html (9.4.2002)

Marshall, C. & Rossman, G. (1989), Designing qualitative research. Sage Publications, Newbury Park.

Marshall, R. & Leaney, P. & Botterel, P. (1999), Enhanced product realisation through modular design: An example of product/process integration. In Proceedings of the SDPS/ASME/IEEE Third World Conference on Integrated Design and Process Technology, Berlin, pp. 143-150.

Marshall, R. & Leaney, P. (1999), A systems engineering approach to product modularity. In Proceedings of the I MECH E Part B Journal of Engineering Manufacture, 1 December 1999, vol. 213, no. 8, pp. 847-851. At http://www.lboro.ac.uk/departments/en/research/cae/res_int/ipps/docs/imebsh1.pdf (9.4.2002)

Martin, A. & Eastman, D. (1996), The user interface design book for the applications programmer. Wiley, Chichester.

Mayhew, D. (1999), The usability engineering lifecycle: A practitioner's handbook for user interface design. Morgan Kaufmann, San Francisco.

Maynard Smith, J. & Szathmáry, E. (1999), The origins of life. Oxford University Press, New York.

McKelvey, W. (1998), Self-Organization, complexity catastrophe and microstate models at the edge of chaos. In J. Baum and W. McKelvey (editors): Variations in Organization Science, pp. 279-307. Sage Publications, Thousand Oaks.

McKelvey, W. (1999), Avoiding complexity catastrophe in coevolutionary pockets: Strategies for rugged landscapes. Organization Science vol. 10, no. 3, pp. 294-321.

Meyer, M. & Seliger, R. (1998), Product platforms in software development. Sloan Management Review, Cambridge, vol. 40, no. 1 pp. 61-74.

Microsoft Corporation (1996), The Microsoft manual of style for technical publications. Microsoft Press, Redmond.

Microsoft Corporation. (1995), The Windows interface guidelines for software design. Microsoft Press, Redmond.

Minasi, M. (1994), Secrets of effective GUI design. Sybex, Alameda.

Mullet, K. & Sano, D. (1995), Designing visual interfaces: Communication oriented techniques. SunSoft Press, Mountain View.

Murray, B. (1999), Coevolution of humans and communication technology. International Symposium for Deep Space Communications and Navigation, September 21-23, 1999. At http://www.gps.caltech.edu/~bcm/HomePage/coev-abstract.html (9.4.2002)

Neelakanta, P. & Deecharoenkul, W. (2000), A complex system characterization of modern telecommunication services. Complex Systems vol. 12, no. 1, pp. 31-69.

Nickerson, R. & Landauer, T. (1997), Human-Computer Interaction: Background and issues. In M. Helander, T. Landauer and P. Prabhu (editors): Handbook of Human-Computer Interaction. Second edition. Elsevier, Amsterdam.

Nielsen, J. (1989), Coordinating user interfaces for consistency. Academic Press, Boston.

Nielsen, J. & Mack, R. (editors) (1994), Usability inspection methods. Wiley & sons, New York.

Nielsen, J. (1993), Usability engineering. Morgan Kaufmann Publishers, San Francisco.

Niiniluoto, I. (1999, 2nd ed.), Johdatus tieteenfilosofiaan. Käsitteen- ja teorianmuodostus. Otava, Helsinki.

Nilsson, M. & Snoad, N. (2000), Error threshold on dynamic fitness-landscapes. PRL, vol. 84 (3 Jan 2000), pp. 191-194. Santa Fe Institute working paper 99-04-030. At http://www.santafe.edu/sfi/publications/wpabstract/199904030 (9.4.2002)

Nitecki, M. (editor) (1983), Coevolution. University of Chicago Press, Chicago.

Norman, D. (1990), The design of everyday things. Doubleday, New York.

Norman, K. (1991), The psychology of menu selection: Designing cognitive control at the human/computer interface. Ablex, Norwood.

Open Software Foundation (1993), OSF/Motif style guide. Prentice Hall, Englewood Cliffs.

Parker, R. (1997), Guide to web content and design: Eight steps to web site success. MIS:Press, New York.

Patton, M. (1990, 2nd ed.), Qualitative evaluation and research methods. Sage Publications, Newbury Park.

Penrose, R. (1989), The emperor's new mind. Oxford University Press, New York.

Poon, L. & Grebogi, C. (1995), Controlling complexity. Physical Review Letters, vol. 75, no. 22, p. 4023.

Proctor, R. & Van Zandt, T. (1994), Human factors in simple and complex systems. Allyn & Bacon, Boston.

Rasmussen, J. & Pejtersen Mark, A. & Goodstein, L. (1994), Cognitive systems engineering. John Wiley & Sons, New York.

Ray, T. (1994), Evolution, complexity, entropy and artificial reality. Physica D 75, pp. 239-263.

Reeves, B. & Nass, C. (1996), The media equation: How people treat computers, television and new media like real people and places. CLSI Publications, Stanford.

Rosenfeld, L. & Morville, P. (1998), Information architecture for the world wide web. O'Reilly & Associates, Cambridge.

Roughgarden, J. (1976), Resource partitioning among competing species – a coevolutionary approach. Theoretical Population Biology. 9, pp. 388-424.

Rubin, J. (1994), Handbook of usability testing: How to plan, design and conduct effective tests. Wiley, New York.

Salvendy, G. (editor) (1997, 2nd edition), Handbook of human factors. Wiley, New York.

Sanchez, R. (1996), Strategic product creation: Managing new interactions of technology, markets, and organizations. European Management Journal, vol. 14, no. 2, pp. 121-138.

Sanders, M. & McCormick, E. (1993, 7th edition), Human factors in engineering and design. McGraw-Hill, New York.

Sano, D. (1996), Designing large-scale web sites: A visual design methodology. Wiley, New York.

Saura, A. (2000), University of Umeå. Personal communication.

Schilling, M. (2000), Toward a general modular systems theory and its application to interfirm product modularity. Academy of Management Review, vol. 25, no. 2, pp. 312-334.

Schneiderman, B. (1998), Designing the user interface: Strategies for Human-Computer Interaction. Addison-Wesley, Reading.

Schumpeter, J. (1934), The theory of economic development. Harvard University Press, Cambridge.

Schuster, P. (1996), How does complexity arise in evolution? – Nature's recipes for mastering scarcity, abundance and unpredictability. Complexity, vol. 2, no. 1, pp. 22-30.

Schwandt, T. (2000) Three epistemological stances for qualitative inquiry: Interpretivism, hermeneutics and social construction. In N. Denzin and Y. Lincoln (editors): Handbook of qualitative research. Sage Publications, Thousand Oaks.

Shipman, R. & Shackleton, M. & Harvey, I. (2000), The use of neutral genotype-phenotype mappings for improved evolutionary search. British Telecom Technology Journal, vol. 18, no. 4, pp. 103-111.

Simonsen, J. & Kensing, F. (1997), Using ethnography in contextual design. Communications of the ACM, vol. 40, no. 7, July.

Singh, K. (1997), The impact of technological complexity and interfirm cooperation on business survival. Academy of Management Journal, vol. 40, no. 2, pp. 339-367.

Stake, R. (1995), The art of case study research. Sage Publications, Thousand Oaks.

Sun Microsystems (1989), OPEN LOOK Graphical user interface application style. Addison-Wesley, Reading.

Tang, V. & Salminen, V. (2001), Towards a theory of complicatedness: Framework for complex systems analysis and design. In 13th International conference on engineering design. August 2001, Glasgow, UK.

Tashakkori, A. & Teddlie, C. (1998), Mixed methodology. Combining qualitative and quantitative approaches. Sage Publications, Thousand Oaks.

TeleManagement Forum (2000), Telecom operations map. 1201 Mt. Kemble Avenue Morristown, NJ 07960 USA. At www.tmforum.org (9.4.2002)

Thomas, P. (editor) (1995), The social and interaction dimensions of human-computer interfaces. Cambridge University Press, Cambridge.

Tinbergen, N. (1951), The study of instinct. Clarendon Press, Oxford.

Tognazzini, B. (1992), Tog on interface. Addison-Wesley, New York.

Tognazzini, B. (1996), Tog on software design. Addison-Wesley, Reading.

Tsypkin, Y. (1971), Adaptation and learning in automatic systems. Academic Press, New York.

Tufte, E. (1997), Visual explanations: Images and quantities, evidence and narrative. Graphics Press, Cheshire.

Turchin, V. (1977), The phenomenon of science. A cybernetic approach to human evolution. Columbia University Press, New York.

UFG Usability first glossary. At http://www.UsabilityFirst.com/glossary/glossary.taf (9.4.2002)

Ulrich, K. (1995), The role of product architecture in the manufacturing firm. Research Policy vol. 24, no. 3 pp. 419-440.

Upadhyayula, S. (2001), Rapid and flexible product development: An analysis of software projects at Hewlett Packard and Agilent. MSc. thesis at the Massachusetts Institute of Technology. At http://web.mit.edu/cipd/documents/thesis_files/SharmaUpadhyayula2001.pdf (9.4.2002)

Vicente, K. (1999), Cognitive work analysis – Toward safe, productive and healthy computer-based work. Lawrence Erlbaum Associates, Mahwah.

Vincenti, W. (1990), What engineers know and how they know it. Baltimore. Johns Hopkins University Press, Baltimore.

Wagner, G. (1996), Homologues, natural kinds, and the evolution of modularity. American Zoologist, 36, pp. 36-43. At http://www.eeb.yale.edu:8001/papers/HomNat/homnat.html (9.4.2002)

Wainer, H. (1997), Visual revelations: Graphical tales of fate and deception from Napoleon Bonaparte to Ross Perot. Copernicus, New York.

Waldrop, M. (1992), Complexity – The emerging science at the edge of order and chaos. Simon & Schuster, New York.

Weinschenk, S. & Jamar, P. & Yeo, S. (1997), GUI design essentials for Windows 95, Windows 31, World Wide Web. Wiley, New York.

Weiss, P. (1951), Ex tempore contribution in the Hixon symposium. In L. Jeffress (editor): Cerebral mechanisms in behaviour - The Hixon symposium. Wiley & Sons, New York.

Wilcox, S. (2001a), Designing out ego, designing in customer preferences. Appliance Manufacturer, March 2001.

Wilcox, S. (2001b), Ethnography as a product development tool. Appliance Manufacturer, July 2001.

Wright, S. (1932), The roles of mutations, inbreeding, crossbreeding, and selection in evolution. In Proceedings of the 6th International Congress on Genetics, Ithaca, 1, pp. 356-366.

Yin, R. (1994, 2nd ed.), Case study research. Design and methods. Sage Publications, Thousand Oaks.

Zetie, C. (1995), Practical user interface design: Making GUIs work. McGraw-Hill, London.

Ziman, J. (editor) (2000), Technological innovation as an evolutionary process. Cambridge University Press, Cambridge.

Appendices

Appendix 1. A fictional story about buying a car

What would it be like to drive a car in a world like this?

- The designers of the manufacturing companies do not drive a car and hence, do not know much about it.
- Different product lines of the manufacturer produce different subsystems of the car and do not release their products simultaneously.
- Different releases of various subsystems are not necessarily compatible with each other.
- Some parts that the customers want are either not produced in the company or are not usable. Consequently, the drivers must produce their own additional equipment or buy it elsewhere to be able to operate their cars.
- Instead of having a simple steering mechanism and 2…3 pedals, there is a complicated "Car Management System", CMS, using a large number of separate tools.
- Many control mechanisms are manual and require many different tools for each task.
- The management system comes separately for an additional price.
- Steering the car is done by manually tuning a large number of low level parameters.

Imagine a guy walking to a car dealer to buy a car.

- *Hello! I would like to buy a car.*
- *Great. Step right in. I will introduce you to our latest models. This is our top of the line model. It has everything you could possibly want.*
- *Oh, good.*
- *Of course, you have to upgrade it occasionally. We develop new features for it all the time. You want to keep up with the development, don't you?*
- *Certainly. But I can get the model with the present features right now, can't I?*
- *Well, yes. We also have many optional features. Would you like to take them too?*
- *Hmm, what kind of optional features?*
- *Ok, I can ask our experts to tell you about the features and how to operate them.*
- *Experts, you say? How many of them?*
- *Oh, 16 or 17 different guys. It's a rather complicated machine and requires a lot of expertise to control.*
- *What?! I just want to drive around with the car. Do I really have to know all that?*

- *To drive… Oh yes…Hmmm, we don't know much about that, but I can assure you that we know how to operate the necessary equipment.*
- *If you don't know much about driving, how have you designed all that 'necessary' equipment?*
- *We get a lot of comments and wishes for new features from our customers. Everything is based on that data.*
- *I see. Just how do you gather it? I presume you have a team of experienced researchers, who spend weeks and weeks in the field to really experience all those problems with the customers, so that they can design comprehensive solutions taking all aspects of the car into consideration. Anyway, that's the way I would do it.*
- *No, we have some people from each product line to have discussions with the customers a few times every year.*
- *Do they go there together?*
- *No, each product line goes separately.*
- *Sounds like pretty fragmented data. How do you expect to be able to produce comprehensive system wide solutions then? I would imagine that your solutions lack system usability.*
- *System what…? We don't do any business with usability. The business is in selling solutions.*
- *Actually, soon you may not do any business what so ever, if some vendor introduces total system usability in their cars. I can tell you for a fact, that I wouldn't buy a car like this, if I had a choice. Unfortunately, there isn't much choice at the moment.*
- *Our business is increasing every year. I don't see any problem in that.*
- *Ok. You mentioned that you get lots of wishes for new features from the customers. Do you mean that you will then produce those features?*
- *Yes. Well, kind of… We screen them first and then decide how many we can produce in the next release. Some of them could also go to a more distant release, but most of them are left out due to lack of resources.*
- *And how long does it take for a customer to get such a feature, if he is lucky, that is?*
- *If it comes in the next release, he might get it even after a couple of years. At least … almost.*
- *I see. And what if the realisation is not good. After a couple of years development without customer feedback that is quite likely. If the customers want to change something in it, then what? Don't tell me, let me guess: They have to wait another two years.*
- *Yes, that is the way we do business.*
- *Let me interpret this. When you offer the improvement, the world has changed so much that the customers don't need that any more. They need something else now. In this modern, fast-changing world, it is impossible to offer up-to-date solutions with that kind of long and inflexible product development process. You are lagging two years behind the customer needs*

all the time. But let's have that talk with the technical experts and see where we stand. I'm interested in seeing how this ends.

………………….

- *…and you can check the speed of the car by activating this measurement, which will produce a measurement result file. Then you can check the statistics in an easy graphical form. You can activate any other measurement in the same manner, if you want to know some other information about the system. I would recommend that you have several measurements going on all the time, because that information is required when you optimise the parameters necessary for controlling the car. For example, you have to know the rpm data in order to change the parameters of the adjustable intake runner system, the fuel injection system, the ignition timing system, intake and exhaust valve opening system, the pollution control system, just to mention a few. And of course, the rpm data is not the only thing you need to know for the parameter optimisation, there are many others, but all of them are readily available. All these things are controlled with different software applications each displaying in its own window.*
- *And how many control parameters are there?*
- *In the current revision about 200. Of course, that increases in every release, when we introduce our new features. There is also a parameter course available from us at very attractive price.*
- *So you don't ever redesign or remove anything. You just add all the new features on top of the old stuff?*
- *Yes, that's the way we do business. By the way, did I mention that all the different subsystem releases are separate? We do not synchronise them.*
- *I'm curious. What does that mean in practice?*
- *It means that whenever you buy a new PPS release, that's short for Power Plant Subsystem, there is no CMS support available. That will come a little later, whenever the product line will get it ready. In the mean time, you have to manage with the new features and parameters on your own. Oh, did I mention that they charge for the CMS system separately? It is not included in the price.*
- *You mean that when I buy a car, I'm actually buying just a part of it and then I have to come up with more money to get the rest of it? If I actually want to use it, that is. Sounds strange to me.*
- *Hmm, yes … If you want to look at it that way…*
- *Why do you have so many different parameters in there? I don't want to hassle with them all the time.*

- *Well, we have a customer who insists on having a control over everything possible and we aim to please.*
- *Only one? You have created all that complexity just to please one customer? And no other customer wants to fool around with all these details? Is that wise thinking?*
- *Well ... It ... Hmm ... I would ... Surely ... Now that you mention it ... It sounds a little odd indeed.*
- *About all those different control applications. Isn't it a little confusing to have so many different control windows open there simultaneously? After a while you don't even remember what you were doing in each window.*
- *Well, complicated systems require complicated control mechanisms.*
- *Do they really? I don't completely agree. By the way, does this abundance of applications even satisfy the needs of the customers? I mean, can they cope with your applications only?*
- *Not exactly. All our customers design additional applications to handle their requirements and also buy some third party applications.*
- *Hmm, I wonder why. Since you don't drive cars, it would be useful for you to do some process research about how people drive them, I imagine. That way you could design those applications to suit better with the actual customer requirements.*
- *Yes, we actually do that. There are some people in the CMS-unit, who have started doing it a while ago.*
- *But there are no representatives from other units with them? Once again, how do you think you can get the comprehensive picture? And how do you expect it to spread through the whole company?*
- *We don't. We reckon it's enough if the CMS guys deal with the operating processes.*
- *Really? By the way, it seems to me that you are stipulating the way that the cars are operated. I claim that if the drivers could decide for themselves how to drive a car, it would be totally different. It would be much easier. What the car drivers need, is system usability, and if some vendor could offer that, the rest would be in bad trouble.*
- *What do you mean by system usability?*
- *Well, it means many things. For example, it means designing the whole system based on comprehensive understanding of user's operating processes, it means reduction of user-observable complexity, it means elimination of unnecessary manual work. To put it short, it means making the use of a complicated system effortless and economical, even fun.*
- *And you think that is possible?*
- *Yes, I definitely think it is possible. More than that, I think it is inevitable. Sooner or later some vendor will do it and the rest will become extinct, if they don't follow.*

Appendix 2. Definition of Systems Theory

Prepared for the Cambridge Dictionary of Philosophy by Francis Heylighen and Cliff Joslyn.

Systems Theory [Including Systems Analysis]: the transdisciplinary study of the abstract *organisation* of phenomena, independent of their substance, type, or spatial or temporal scale of existence. It investigates both the principles common to all complex entities, and the (usually mathematical) *models* which can be used to describe them.

Systems theory was proposed in the 1940's by the biologist Ludwig von Bertalanffy (anthology: General Systems Theory, 1968), and furthered by Ross Ashby (Introduction to Cybernetics, 1956). von Bertalanffy was both reacting against *reductionism* and attempting to revive the *unity of science*. He emphasised that real systems are open to, and interact with, their environments, and that they can acquire qualitatively new properties through *emergence*, resulting in continual *evolution*. Rather than reducing an entity (e.g. the human body) to the properties of its parts or elements (e.g. organs or cells), systems theory focuses on the arrangement of and *relations* between the parts which connect them into a whole (cf. *holism*). This particular *organisation* determines a *system*, which is independent of the concrete substance of the elements (e.g. particles, cells, transistors, people, etc). Thus, the same concepts and principles of organisation underlie the different disciplines (physics, biology, technology, sociology, etc.), providing a basis for their unification. Systems concepts include: system-environment *boundary, input, output, process, state, hierarchy, goal-directedness*, and *information*.

The developments of systems theory are diverse (Klir, Facets of Systems Science, 1991), including conceptual foundations and philosophy (e.g. the philosophies of Bunge, Bahm and Laszlo); mathematical modelling and *information theory* (e.g. the work of Mesarovic and Klir); and practical applications. Mathematical systems theory arose from the development of isomorphism between the models of electrical circuits and other systems. Applications include engineering, computing, ecology, management, and family psychotherapy. Systems analysis, developed independently of systems theory, applies systems principles to aid a decision-maker with problems of identifying, reconstructing, optimising, and controlling a system (usually a socio-technical organisation), while taking into account multiple objectives, constraints and resources. It aims to specify possible courses of action, together with their risks, costs and benefits. Systems theory is closely connected to *cybernetics*, and also to *system dynamics*, which model changes in a *network* of coupled variables (e.g. the "world dynamics" models of Jay Forrester and the Club of Rome). Related ideas are used in the emerging "sciences of *complexity*", studying *selforganisation* and heterogeneous networks of interacting actors, and associated domains such as *far-from-equilibrium thermodynamics, chaotic dynamics, artificial life, artificial intelligence, neural networks*, and computer modelling and simulation.

Appendix 3. Operator A's process improvements after the research

There has been some development in the processes of Operator A after the initial field work. We briefly describe those improvements in the following.

Cell creation

When the radio network planner designs the network plan, he imports the new plan to the NMS-database. At this stage, there is no longer a need to manually create the views for the new BTSs, as they are automatically generated into the TLUI. However, the NMS-user in the network control room checks (and in practice corrects) the view with an NMS-tool. In this new procedure, there is no need to create the BTS parameters and neighbour cell definitions either, since the parameters and neighbour definitions are already in the NMS-database after the plan import.

The new cell creation/neighbour setting task would look like this:

1. Import the network plan in NMS (Network planner)
2. Check and correct the view (NMS-user)
3. Create and store the links (NMS-user)
4. Send the plan to the network (NMS-user)
5. Give the network planners the right to edit the data (NMS-user)

Sometimes it happens that the network planner forgets to do this plan import and hence the data is not in the NMS-database. In this case, the NMS-users must revert to the old way and manually create the views, set the parameters and neighbours.

Hi-Tech Inc. has created a graphical adjacency manager for the NMS, but the users do not consider it much better than the old NMS-tool. Some users deploy the new one, some the old one. The policy on the supplier side seems unchanged in the sense that the number of applications is increased all the time.

Assessment

The operator has tackled the view creation/parameter setting dependency discovered in the research. The K-factor between the tasks of different work groups has been reduced by eliminating the mentioned dependency. In addition, the operating effort has been reduced by increased automation. Although, everything is not going smoothly yet (e.g. the views must be manually corrected), this is an improvement.

The supplier could follow the metasystem transition principle and do a major restructurisation of the tool set, when the number of tools becomes too high. In such a situation, the functionality should be designed to follow the recommended operating processes. The processes would be designed first based on the operator process research (see chapter 8.3.2).

Neighbour definition

There is also an improvement in the neighbour setting/modifying. Hi-Tech Inc. has created a new tool, which resembles the idea suggested in the improvement suggestions of the neighbour setting process. The neighbours can now be set by using a graphical tool. There is a view, where the wanted BTSs can be picked up one by one and there is also a selection whether the definitions should be one way or two way. The only difference compared to the suggestion is that in this tool there is no possibility to select many BTSs at once. In the suggestion, this was possible with the aid of a user defined circle. As a minor detail, the direction default is wrong. The default should be bi-directional definitions, as this is the case practically always.

Setting the GPRS parameters in the commissioning process

For GPRS, there are some new parameters, which must be defined for every cell and every TRX. This is done by using MML-commands, since presently the graphical tool in NMS cannot quite manage this task.

References in index order

---

[1] ISO/IEC 9126-1 (2001), Software engineering - Product quality. Part 1: Quality model. At http://www.iso.ch/iso/en/ISOOnline.frontpage (9.4.2002)

[2] ISO 9241-11 (1998), Ergonomic requirements for office work with visual display terminals (VDTs). Part 11: Guidance on usability. At http://www.iso.ch/iso/en/ISOOnline.frontpage (9.4.2002)

[3] Adams, E. & Willets, K. (1996), The lean communications provider. Surviving the shakeout through service management excellence. McGraw-Hill, New York.

[4] Alvesson, M. & Sköldberg, K (2000), Reflexive methodology. New vistas for qualitative research. Sage Publications, Thousand Oaks.

[5] Geertz, C. (1979) From the native's point of view: On the nature of anthropological understanding. In P. Rabinow & W. Sullivan (editors), Interpretive social science: A reader, pp. 225-241. University of California Press, Berkeley.

[6] Schwandt, T. (2000) Three epistemological stances for qualitative inquiry: Interpretivism, hermeneutics and social construction. In N. Denzin and Y. Lincoln (editors), Handbook of qualitative research. Sage Publications, Thousand Oaks.

[7] Simonsen, J. & Kensing, F. (1997), Using ethnography in contextual design. Communications of the ACM, vol. 40, no. 7, July.

[8] Blomberg, J. & Giacomi, J. & Mosher, A. & Swenton-Hall, P. (1993), Ethnographic field methods and their relation to design. In D. Schuler and A. Namioka (editors): Participatory design: Principles and practices, pp. 123-155. Lawrence Erlbaum Associates, Hillsdale.

[9] Bohm, D. (1980), Wholeness and the implicate order. Routledge, London.

[10] Johnson, J. (2002), In-depth interviewing. In J. Gubrium and J. Holstein (editors): Handbook of interview research: Context and method. Sage Publications, Thousand Oaks.

[11] Niiniluoto, I. (1999, 2nd ed.), Johdatus tieteenfilosofiaan. Käsitteen- ja teorianmuodostus. Otava, Helsinki.

[12] Yin, R. (1994, 2nd ed.), Case study research. Design and methods. Sage Publications, Thousand Oaks.

[13] ETSI specification ETR095 (1993), Human factors (HF); Guide for usability evaluations of telecommunications systems and services. European Telecommunications Standards Institute. At http://www.etsi.org/ (9.4.2002)

[14] UFG Usability first glossary. At http://www.UsabilityFirst.com/glossary/glossary.taf (9.4.2002)

[15] Hom, J. (1996), Usability methods toolbox. At http://www.best.com/~jthom/usability/ (9.4.2002)

[16] Stake, R. (1995), The art of case study research. Sage Publications, Thousand Oaks.

[17] Gubrium, J. & Holstein, J. (2002), From the individual interview to the interview society. In J. Gubrium and J. Holstein (editors): Handbook of interview research: Context and method. Sage Publications, Thousand Oaks.

[18] Patton, M. (1990, 2nd ed.), Qualitative evaluation and research methods. Sage Publications, Newbury Park..

[19] Tashakkori, A. & Teddlie, C. (1998), Mixed methodology. Combining qualitative and quantitative approaches. Sage Publications, Thousand Oaks.

[20] Rasmussen, J. & Pejtersen Mark, A. & Goodstein, L. (1994), Cognitive systems engineering. John Wiley & Sons, New York..

[21] Nickerson, R. & Landauer, T. (1997), Human-Computer Interaction: Background and issues. In M. Helander, T. Landauer and P. Prabhu (editors): Handbook of Human-Computer Interaction. Second edition. Elsevier, Amsterdam.

[22] Brodie, E. (1997a), Evolution course. Course material. Department of Biology. University of Indiana. At http://www.bio.indiana.edu/courses/L318-brodie/L318/L318lect16-fitness.html (9.4.2002)

---

[23] Britannica.com Inc (2001), Encyclopedia Britannica. At http://www.britannica.com (9.4.2002)

[24] Encarta Reference. Microsoft encyclopedia at http://encarta.msn.com/reference (9.4.2002)

[25] Colby, C. (1996), Introduction to evolutionary biology. At http://www.talkorigins.org/faqs/faq-intro-to-biology.html (9.4.2002)

[26] ACM SIGCHI (1992), Curricula for Human-Computer Interaction. At http://www.acm.org/sigchi/cdg (9.4.2002)

[27] Carey, J. (editor) (1997), Human factors in information systems: The relationship between user interface design and human performance. Ablex, Greenwich.

[28] Chapanis, A. (1996), Human factors in system engineering. Wiley, New York.

[29] Coe, M. (1996), Human factors for technical communicators. Wiley, New York.

[30] Norman, D. (1990), The design of everyday things. Doubleday, New York.

[31] Salvendy, G. (editor) (1997, 2nd edition), Handbook of human factors. Wiley, New York.

[32] Sanders, M. & McCormick, E. (1993, 7th edition), Human factors in engineering and design. McGraw-Hill, New York.

[33] Baecker, R. & Grudin, J. & Buxton, W. & Greenberg, S. (editors) (1995, 2nd edition), Readings in Human-Computer Interaction: Toward the year 2000. Morgan Kaufmann Publishers, San Francisco.

[34] Casey, S. (1998, 2nd edition), Set phasers on stun --- And other true tales of design, technology and human error. Aegean Publishing Company, Santa Barbara.

[35] Dix, A. & Finlay, J. & Abowd, G. & Beale, R. (1998, 2nd edition), Human-Computer Interaction. Prentice Hall Europe, London.

[36] Helander, M. & Landauer, T. & Prabhu, P. (editors) (1997), Handbook of Human-Computer Interaction, second edition. North-Holland, Amsterdam.

[37] Karat, J. (editor) (1991), Taking software design seriously: Practical techniques for Human-Computer Interaction design. Academic Press, Boston.

[38] Schneiderman, B. (1998), Designing the user interface: Strategies for Human-Computer Interaction. Addison-Wesley, Reading.

[39] Card, S. & Moran, T. & Newell, A. (1983), The psychology of human-computer interaction. Lawrence Erlbaum Associates, Hillsdale.

[40] Gardiner, M. & Christie, B. (editors) (1987), Applying cognitive psychology to user-interface design. Wiley & Sons, Chichester.

[41] Norman, K. (1991), The psychology of menu selection: Designing cognitive control at the human/computer interface. Ablex, Norwood.

[42] Proctor, R. & Van Zandt, T. (1994), Human factors in simple and complex systems. Allyn & Bacon, Boston.

[43] Jirotka, M. & Goguen, P. (editors) (1994), Requirements engineering --- Social and technical issues. Academic Press, London.

[44] Lansdale, M. & Ormerod, T. (1994), Understanding interfaces: A handbook of human-computer dialogue. Academic Press, London.

[45] Reeves, B. & Nass, C. (1996), The media equation: How people treat computers, television and new media like real people and places. CLSI Publications, Stanford.

[46] Thomas, P. (editor) (1995), The social and interaction dimensions of human-computer interfaces. Cambridge University Press, Cambridge.

[47] Arlov, L. (1997), GUI Design for dummies. IDG Books, Foster city.

[48] Borenstein, N. (1994), Programming as if people mattered: Friendly programs, Software engineering and other noble delutions. Princeton University Press, Princeton.

[49] Fowler, S. (1998), GUI design handbook. McGraw-Hill, New York.

[50] Marcus, A. & Smilonich, N. & Thompson, L. (1995), The cross-GUI handbook. Addison-Wesley, Reading.

[51] Minasi, M. (1994), Secrets of effective GUI design. Sybex, Alameda.

[52] Tognazzini, B. (1992), Tog on interface. Addison-Wesley, New York.

---

[53] Weinschenk, S. & Jamar, P. & Yeo, S. (1997), GUI design essentials for Windows 95, Windows 31, World Wide Web. Wiley, New York.

[54] Zetie, C. (1995), Practical user interface design: Making GUIs work. McGraw-Hill, London.

[55] Browne, D. (1994), STUDIO: Structured user-interface design for interaction optimisation. Prentice Hall, New York.

[56] Collins, D. (1995), Designing object-oriented user interfaces. Benjamin/Cummings Publishing, Redwood City.

[57] Cooper, A. (1995), About face: The essentials of user interface design. IDG Books Worldwide, Foster City.

[58] Howlett, V. (1996), Visual interface design for Windows. Wiley, New York.

[59] Mandel, T. (1997), The elements of user interface design. Wiley, New York.

[60] Martin, A. & Eastman, D. (1996), The user interface design book for the applications programmer. Wiley, Chichester.

[61] Mayhew, D. (1999), The usability engineering lifecycle: A practitioner's handbook for user interface design. Morgan Kaufmann, San Francisco.

[62] Microsoft Corporation. (1995), The Windows interface guidelines for software design. Microsoft Press, Redmond.

[63] Nielsen, J. (1989), Coordinating user interfaces for consistency. Academic Press, Boston.

[64] Horton, W. (1992), Illustrating computer documentation: The art of presenting information graphically on paper and online. Wiley, New York.

[65] Marcus, A. (1991), Graphic design for electronic documents and user interfaces. Addison Wesley, New York.

[66] Mullet, K. & Sano, D. (1995), Designing visual interfaces: Communication oriented techniques. SunSoft Press, Mountain View.

[67] Tufte, E. (1997), Visual explanations: Images and quantities, evidence and narrative. Graphics Press, Cheshire.

[68] Wainer, H. (1997), Visual revelations: Graphical tales of fate and deception from Napoleon Bonaparte to Ross Perot. Copernicus, New York.

[69] Horton, W. (1994), The icon book: Visual symbols for computer systems and documentation. Wiley, New York.

[70] Albers, J. (1987), Interaction of color. Yale University Press, New Haven.

[71] Bias, R. & Mayhew, D. (editors) (1994), Cost-justifying usability. Academic Press, Boston.

[72] Dumas, J. & Redish, J. (1999, revised edition), A practical guide to usability testing. Intellect Ltd, Bristol.

[73] Lindgaard, G. (1994), Usability testing and system evaluation: A guide for designing useful computer systems. Chapman & Hall, London.

[74] Nielsen, J. & Mack, R. (editors) (1994), Usability inspection methods. Wiley & sons, New York.

[75] Rubin, J. (1994), Handbook of usability testing: How to plan, design and conduct effective tests. Wiley, New York.

[76] Beyer, H. & Holtzblatt, K. (1998), Contextual design: Defining customer-centered systems. Morgan Kaufmann, San Francisco.

[77] Greenbaum, J. & Kyng, M. (editors) (1991), Design at work: Cooperative design of computer systems. Lawrence Erlbaum, Hillsdale.

[78] Kyng, M. & Mathiassen, L.(editors) (1997), Computers and design in context. MIT Press, Cambridge.

[79] Landauer, T. (1995), The trouble with computers: Usefulness, usability, and productivity. MIT Press, Cambridge.

[80] Tognazzini, B. (1996), Tog on software design. Addison-Wesley, Reading.

[81] Apple Computer (1992), Macintosh human interface guidelines. Addison-Wesley, Reading.

[82] Fowler, S. & Stanwick, V. (1995), The GUI style guide. AP professional, Boston.

---

[83] Microsoft Corporation (1996), The Microsoft manual of style for technical publications. Microsoft Press, Redmond.

[84] Open Software Foundation (1993), OSF/Motif style guide. Prentice Hall, Englewood Cliffs.

[85] Sun Microsystems (1989), OPEN LOOK Graphical user interface application style. Addison-Wesley, Reading.

[86] Bain, S. & Gray, D. (1996), Looking good online: The ultimate resource to creating effective web designs. Ventana, Research Triangle Park.

[87] Fosythe, C. & Grose, E. & Ratner, J. (editors) (1998), Human factors and web development paperback. Lawrence Erlbaum, Mahwah.

[88] Parker, R. (1997), Guide to web content and design: Eight steps to web site success. MIS:Press, New York.

[89] Rosenfeld, L. & Morville, P. (1998), Information architecture for the world wide web. O'Reilly & Associates, Cambridge.

[90] Sano, D. (1996), Designing large-scale web sites: A visual design methodology. Wiley, New York.

[91] Bevan, N. (1995a), Measuring usability as quality of use. Journal of Software Quality, Issue 4, pp. 115-140.

[92] Nielsen, J. (1993), Usability engineering. Morgan Kaufmann Publishers, San Francisco.

[93] Keinonen, T. (1998), One-dimensional usability. University of Art and Design Helsinki, UIAH Publications, Helsinki.

[94] Waldrop, M. (1992), Complexity - The emerging science at the edge of order and chaos. Simon & Schuster, New York.

[95] Bertalanffy, L. von (1968, revised edition 1984), General system theory. George Braziller Inc, New York.

[96] Kaneko, K. & Tsuda, I. (1994), Constructive complexity and artificial reality: An introduction. Physica, D 75, pp. 1-10.

[97] Poon, L. & Greborgi, C. (1995), Controlling complexity. Physical Review Letters, vol. 75, no. 22, p. 4023.

[98] Heylighen, F. (1994), Fitness as a default: The evolutionary basis for cognitive complexity reduction. In R. Trappl (editor): Cybernetics and Systems, World Science, Singapore, pp.1595-1602.

[99] Heylighen, F. (1999), The growth of structural and functional complexity during evolution. In F. Heylighen & J. Bollen & A. Riegler (editors): The evolution of complexity, Kluwer Academic, Dordrecht, pp. 17-44. At http://pespmc1.vub.ac.be/papers/ComplexityGrowth.html (9.4.2002)

[100] Turchin, V. (1977), The phenomenon of science. A cybernetic approach to human evolution. Columbia University Press, New York.

[101] Schuster, P. (1996), How does complexity arise in evolution? – Nature's recipes for mastering scarcity, abundance and unpredictability. Complexity, vol. 2, no. 1, pp. 22-30.

[102] Wagner, G. (1996), Homologues, natural kinds, and the evolution of modularity. American Zoologist, 36, pp. 36-43. At http://www.eeb.yale.edu:8001/papers/HomNat/homnat.html (9.4.2002)

[103] Bonner, J. (1988), The Evolution of complexity. Princeton University Press, Princeton.

[104] Marshall, R. & Leaney, P. & Botterel, P. (1999), Enhanced product realisation through modular design: An example of product/process integration. In Proceedings of the SDPS/ASME/IEEE Third World Conference on Integrated Design and Process Technology, Berlin, Germany, pp. 143-150.

[105] Marshall, R. & Leaney, P. (1999), A systems engineering approach to product modularity. In Proceedings of the I MECH E Part B Journal of Engineering Manufacture, 1 December 1999, vol. 213, no. 8, pp. 847-851. At http://www.lboro.ac.uk/departments/en/research/cae/res_int/ipps/docs/imebsh1.pdf (9.4.2002)

[106] Weiss, P. (1951), Ex tempore contribution in the Hixon symposium. In L. Jeffress (editor), Cerebral mechanisms in behaviour - The Hixon symposium. Wiley & Sons, New York.

[107] Koestler, A. (1964), The act of creation. Penguin Books, London.

---

[108] Markoff, J. (2001), Software's next leap is out of the box. New York Times, June 13, 2001. At http://www.nytimes.com/2001/06/13/technology/13MARK.html (9.4.2002)

[109] Jeffress, L. (editor), (1951), Cerebral mechanisms in behaviour - The Hixon symposium. Wiley & Sons, New York.

[110] Tinbergen, N. (1951), The study of instinct. Clarendon Press, Oxford.

[111] Gell-Mann, M. (1995), What is complexity? Complexity vol. 1, no. 1, pp. 16-19.

[112] Kauffman, S. (2000), Investigations. Oxford University Press, New York.

[113] Crutchfield, J. & Nimwegen, E. (1999), The evolutionary unfolding of complexity. Santa Fe Institute working paper 99-02-015. At http://www.santafe.edu/sfi/publications/wpabstract/199902015 (9.4.2002)

[114] Gell-Mann, M. (1994), The Quark and the Jaguar – Adventures in the simple and the complex. Little, Brown and Company, London.

[115] Tang, V. & Salminen, V. (2001), Towards a theory of complicatedness: Framework for complex systems analysis and design. In 13th International conference on engineering design. August 2001, Glasgow, UK.

[116] Vicente, K. (1999), Cognitive work analysis – Toward safe, productive and healthy computer-based work. Lawrence Erlbaum Associates, London.

[117] Gell-Mann, M. & Lloyd, S. (1996), Information measures, effective complexity, and total information. Complexity, vol. 2, no. 1, pp. 44-52.

[118] Craik, K. (1943), The nature of explanation. Cambridge University Press, Cambridge.

[119] Edmonds, B. (1995), What is complexity? – The philosophy of complexity *per se* with application to some examples in evolution. In F. Heylighen and D. Aerts (editors): The Evolution of Complexity, Kluwer Academic, Dordrecht. At http://www.cpm.mmu.ac.uk/~bruce/evolcomp/evolcomp_1.html (9.4.2002)

[120] Arthur, W. (1993), Why do things become more complex? Scientific American, May 1993, p 144.

[121] Tsypkin, Y. (1971), Adaptation and learning in automatic systems. Academic Press, New York.

[122] Gell-Mann, M. (2002), Plectics: The study of simplicity and complexity. Europhysics News, vol. 33, no. 1, pp. 17-20.

[123] Bartlett, F. (1961), Remembering. Cambridge University Press, Cambridge.

[124] Holland, J. (1995), Hidden order – How adaptation builds complexity. Addison-Wesley, Reading.

[125] Wright, S. (1932), The roles of mutations, inbreeding, crossbreeding, and selection in evolution. In Proceedings of the 6th International Congress on Genetics, Ithaca, 1, pp. 356-366.

[126] Kauffman, S. (1992), The origins of order – Self-organization and selection in evolution. Oxford University Press, New York.

[127] Vincenti, W. (1990), What engineers know and how they know it. Baltimore. Johns Hopkins University Press, Baltimore.

[128] Ziman, J. (editor), (2000), Technological innovation as an evolutionary process. Cambridge University Press, Cambridge.

[129] Fleming, L. & Sorenson, O. (1999), Technology as a complex adaptive system: Evidence from patent data. Forthcoming in Research Policy. At http://www.anderson.ucla.edu/faculty/olav.sorenson/fs99.pdf (9.4.2002)

[130] Fleming, L. & Sorenson. O. (2001), Science, Technological interdependence, and invention. Harvard Business School Working paper.

[131] Shipman, R. & Shackleton, M. & Harvey, I. (2000), The use of neutral genotype-phenotype mappings for improved evolutionary search. British Telecom Technology Journal, vol. 18, no. 4, pp. 103-111.

[132] Kauffman, S. & Lobo, J. & Macready, W. (1998), Optimal search on a technology landscape. Santa Fe Institute working paper 98-10-091. At http://www.santafe.edu/sfi/publications/wpabstract/199810091 (9.4.2002)

---

[133] Kauffman, S. & Macready, W. (1995), Technological evolution and adaptive organizations. Complexity, vol. 1, no. 2, pp. 26-43.

[134] Kauffman, S. (1995), At home in the universe – The search for laws of complexity. Penguin books, London.

[135] Saura, A. (2000), University of Umeå. Personal communication.

[136] Eigen, M. & Schuster, P. (1979), The Hypercycle - A principle of natural self-organisation. Springer-Verlag, Berlin.

[137] Maynard Smith, J. & Szathmáry, E. (1999), The origins of life. Oxford University Press, New York.

[138] Singh, K. (1997), The impact of technological complexity and interfirm cooperation on business survival. Academy of Management Journal, vol. 40, no. 2, pp. 339-367.

[139] McKelvey, W. (1998), Self-Organization, complexity catastrophe and microstate models at the edge of chaos. In J. Baum & W. McKelvey (editors): Variations in Organization Science, pp. 279-307. Sage Publications, Thousand Oaks.

[140] Funes, P. & Sklar, E. & Juillé, H. & Pollack, J. (1998), Animal-animat coevolution: Using the animal population as fitness function. In R. Pfeifer et. al. (editors): From animals to animats 5: Proceedings of the fifth international conference on simulation of adaptive behavior. MIT Press. Pp. 525-533. At http://www.demo.cs.brandeis.edu/papers/ (9.4.2002)

[141] Ehrlich, P. & Raven, P. (1964), Butterflies and plants: A study in coevolution. Evolution 18, pp. 586-608.

[142] Course material. Brown University. Division of Biology and Medicine. Department of Ecology and Evolutionary Biology. At http://biomed.brown.edu/Courses/BIO48/27.Coevolution.HTML (9.4.2002)

[143] Brodie, E. (1997b), Evolution course. Course material. University of Indiana. Department of Biology. At http://www.bio.indiana.edu/courses/L318-brodie/L318/L318lect27coevolution.html (9.4.2002)

[144] Gitzendanner, M. (2000), Principles of organic evolution. Course material. Washington State University. Ecology and Evolutionary Biology. At http://www.wsu.edu/~mgitzend/evolution/Notes15b/Notes15b.pdf (9.4.2002)

[145] Eskikaya, B. (1998), A glimpse at artificial coevolution. Course material. University of Sussex at Brighton. At http://www.cogs.susx.ac.uk/lab/nlp/gazdar/teach/atc/1998/talksnode25.html (9.4.2002)

[146] Nitecki, M. (editor) (1983), Coevolution. University of Chicago Press, Chicago.

[147] Baum, J. & Singh, J. (1994), Organization-environment coevolution. In J. Singh (editor): Organizational evolution: New directions. pp. 379-401. Sage Publications, Newbury Park.

[148] Murray, B. (1999), Coevolution of humans and communication technology. International Symposium for Deep Space Communications and Navigation, September 21 through 23, 1999. At http://www.gps.caltech.edu/~bcm/HomePage/coev-abstract.html (9.4.2002)

[149] Beinhocker, E. (1997), Strategy at the edge of chaos. The McKinsey Quarterly, no. 1, pp. 24-39.

[150] Roughgarden, J. (1976), Resource partitioning among competing species – a coevolutionary approach. Theoretical Population Biology. 9, pp. 388-424.

[151] Ray, T. (1994), Evolution, complexity, entropy and artificial reality. Physica D 75, pp. 239-263.

[152] Nilsson, M. & Snoad, N. (2000), Error threshold on dynamic fitness-landscapes. PRL, vol. 84 (3 Jan 2000), pp. 191-194. Santa Fe Institute working paper 99-04-030 at http://www.santafe.edu/sfi/publications/wpabstract/199904030 (9.4.2002)

[153] Schumpeter, J. (1934), The theory of economic development. Harvard University Press, Cambridge.

[154] Anderson, P. & Tushman, M. (1990), Technological discontinuities and dominant design: A cyclical model of technological change. Administrative Science Quarterly 35, pp. 604-633.

[155] McKelvey, W. (1999), Avoiding complexity catastrophe in coevolutionary pockets: Strategies for rugged landscapes. Organization Science vol. 10, no. 3, pp. 294-321.

---

[156] Bendett, R. & Neelakanta P. (2000), A relative complexity metric for decision-theoretic applications in complex systems. Complex Systems vol. 12, no. 3, pp. 281- 295.
[157] Neelakanta, P. & Deecharoenkul, W. (2000), A complex system characterization of modern telecommunication services. Complex Systems vol. 12, no. 1, pp. 31-69.
[158] Cooper, R. (1990), Stage-Gate Systems: A New Tool for Managing New Products. Business Horizons May-June 1990, pp. 44-54.
[159] MacCormack, A. (2000), Towards Contingent Model of the New Product Development Process: A Comparative Empirical Study. A working paper. Harvard Business School, Boston.
[160] TeleManagement Forum (2000), Telecom operations map. 1201 Mt. Kemble Avenue Morristown, NJ 07960 USA. At www.tmforum.org (9.4.2002)
[161] Hi-Tech Inc. FM (1998), NMS For Managing Cellular Networks: Fault Management Basic Operating Principles and Procedures. Not public.
[162] Hi-Tech Inc. CM (1998), NMS For Managing Cellular Networks: Configuration Management Basic Operating Principles and Procedures. Not public
[163] Hi-Tech Inc. PM (1998), NMS For Managing Cellular Networks: Performance Management Basic Operating Principles and Procedures. Not public.
[164] ETSI specification ETS 300 615 (1996), Digital cellular telecommunications system (Phase 2) (GSM): Performance data measurements (GSM 12.04), European Telecommunications Standards Institute. At http://www.etsi.org/ (9.4.2002)
[165] De Vos, G. (1997), The Alcatel SDH network release concept. Alcatel Telecommunications Review, 1st Quarter 1997, pp. 19-23.
[166] Lucas, M. & Meech, F. & Purcell, C. (1997), Re-engineering a complex network interface. In M. Smith & G. Salvendy & R. Koubek (editors): Design of computing systems: Social and ergonomic considerations. Proceedings of the 7th international conference in Human-Computer Interaction, vol. 2. San Francisco, USA. Elsevier, Amsterdam.
[167] Wilcox, S. (2001a), Designing out ego, designing in customer preferences. Appliance Manufacturer, March 2001.
[168] ETSI specification EG 201 472 (2000), Human factors (HF): Usability evaluation for the design of telecommunication systems, services and terminals. European Telecommunications Standards Institute. At http://www.etsi.org/ (9.4.2002)
[169] Leonard, D. & Rayport, J. (1997), Spark innovation through empathic design. Harvard Business Review, November-December 1997.
[170] Lester, R. & Piore, M. & Malek, K. (1998), Interpretive management: What general managers can learn from designers. Harvard Business Review, March-April 1998.
[171] Ashby, W. (1958), Requisite variety and implications for control of complex systems. Cybernetica 1, pp. 83-99.
[172] Christensen, W. & Collier, J. & Hooker, C. (2000), Adaptiveness and adaptation: A new autonomy-theoretic analysis. The University of Newcastle – Australia. At http://www.newcastle.edu.au/centre/casrg/members/christensenwayne/Ch3-Adaptiveness.pdf (9.4.2002)
[173] Ulrich, K. (1995), The role of product architecture in the manufacturing firm. Research Policy vol. 24, no. 3 pp. 419-440.
[174] Baldwin, C. & Clark, K. (2000), Design rules: The power of modularity. MIT Press, Cambridge.
[175] Baldwin, C. & Clark, K. (1997), Managing in an age of modularity. Harvard Business Review, vol. 75, no. 5, pp. 84-93.
[176] Schilling, M. (2000), Toward a general modular systems theory and its application to interfirm product modularity. Academy of Management Review, vol. 25, no. 2, pp. 312-334.
[177] Sanchez, R. (1996), Strategic product creation: Managing new interactions of technology, markets, and organizations. European Management Journal, vol. 14, no. 2, pp. 121-138.

---

[178] Christensen, W. & Hooker, C. (1999), Autonomy and the emergence of intelligence: Organised interactive construction. The University of Newcastle – Australia. At http://www.newcastle.edu.au/centre/casrg/publications/AUTO-NPO.pdf (9.4.2002)

[179] Upadhyayula, S. (2001), Rapid and flexible product development: An analysis of software projects at Hewlett Packard and Agilent. MSc. thesis at the Massachusetts Institute of Technology. At http://web.mit.edu/cipd/documents/thesis_files/SharmaUpadhyayula2001.pdf (9.4.2002)

[180] Dahan, E. & Hauser, R. (2000), The virtual customer: Communication, conceptualization, and computation. MIT Sloan School of Management. At http://mitsloan.mit.edu/vc/Pages/papers.html (9.4.2002)

[181] Dragut, A. & Bertrand, J. (2001), Time and resource characteristics of radical new product development (NPD) projects and their dynamic control. In Proceedings of the 12th annual conference of the production and operations management society, March 30-April 2, 2001, Orlando.

[182] Iansiti, M. & MacCormack A. (1997), Developing product on internet time. Harvard Business Review, vol. 75, no. 5, pp. 108-117.

[183] Hamel, G. (2001), Revolution vs. evolution: You need both. Harvard Business Review, vol. 79, no. 5, pp. 150-153.

[184] Brown, S. & Eisenhardt, K. (1998), Competing on the edge – Strategy as structured chaos. Harvard Business School Press, Boston.

[185] Crabtree, A. & Nichols, D. & O'Brien, J. & Rouncefield, M. & Twidale, M. (2000), Ethnomethodologically informed ethnography and information system design. Journal of the American society for information science. May 2000 pp. 666-682.

[186] Wilcox, S. (2001b), Ethnography as a product development tool. Appliance Manufacturer, July 2001.

[187] Cusumano, M. & Yoffie, D. (1998), Competing on Internet time: Lessons from Netscape and its battle with Microsoft. Free Press/Simon & Schuster, New York.

[188] Eisenhardt, K. (1999), Strategy as strategic decision making. Sloan Management Review, vol. 40, no. 3.

[189] Kadoda, G. (2000), A cognitive dimensions view of the differences between designers and users of theorem proving assistants. In A. Blackwell & E. Bilotta (editors): Proceedings of 12th workshop of the psychology of programming interest group, Cozenza, April 2000, pp. 33-44.

[190] Madsen, K. & Borgholm, T. (1999), Cooperative Usability Practices. Communications of the ACM vol. 42, no. 5.

[191] Buur, J. & Bodker, S. (2000), From usability lab to design collaboratorium: Reframing usability practice. Designing Interactive Systems Conference (DIS 2000), ACM Press, New York.

[192] Hansen, M. (1999), The search-transfer problem: The role of weak ties in sharing knowledge across organisation subunits. Administrative Science Quarterly, March 1999.

[193] Cliffe, S. (1998), Knowledge management: The well-connected business. Harvard Business Review (Briefings from the editors). July-August 1998.

[194] Meyer, M. & Seliger, R. (1998), Product platforms in software development. Sloan Management Review, Cambridge, vol. 40, no. 1 pp. 61-74.

[195] Maguire, S. & McKelvey, W. (1999), Complexity and management: Moving from fad to firm foundations. Emergence, vol. 1, no. 2.

[196] Penrose, R. (1989), The emperor's new mind. Oxford University Press, New York.

[197] Lincoln, Y. & Guba, E. (1985), Naturalistic inquiry. Sage Publications. Newbury Park.

[198] Alasuutari, P. (1994), Laadullinen tutkimus. Vastapaino, Tampere.

[199] Marshall, C. & Rossman, G. (1989), Designing qualitative research. Sage Publications, Newbury Park.

[200] Bevan, N. (1995b), Usability is Quality of Use. In Y. Anzai & K. Ogawa (editors): Proceedings of the 6th international conference on human computer interaction, Yokohama, July 1995. Elsevier, Amsterdam.

---

[201] Kemmis, S. & McTaggart, R. (2000), Participatory action research. In N. Denzin & Y. Lincoln (editors): Handbook of qualitative research, 2nd edition, pp. 567-605. Sage Publications, Thousand Oaks.