Facilitating
Computer Supported Cooperative Work
with
Socio-Technical Self-Descriptions

Dissertation

zur Erlangung des Grades eines
Doktors der Naturwissenschaften
der Universität Dortmund
am Fachbereich Informatik

von

Gabriele Kunau

Dortmund
2006

Tag der mündlichen Prüfung:

10. Februar 2006

Dekan:

Prof. Dr. Bernhard Steffen

Gutachter/innen:

Prof. Dr.-Ing. Thomas Herrmann

Prof. Dr. Gabriele Kern-Isberner

Prof. Dr. Paul Dourish

Table of Contents

GINKGO BILOBA

Dieses Baumes Blatt, der von Osten
Meinem Garten anvertraut,
Gibt geheimen Sinn zu kosten,
Wie's den Wissenden erbaut.

Ist es ein lebendig Wesen,
Das sich in sich selbst getrennt?
Sind es zwei, die sich erlesen,
Daß man sie als eines kennt?

Solche Fragen zu erwidern
Fand ich wohl den rechten Sinn.
Spürst du nicht an meinen Liedern,
Daß ich eins und doppelt bin ?

GINKGO BILOBA

This tree's leaf which here the East
In my garden propagates,
On its secret sense we feast,
Such as sages elevates.

Is it one but being single
Which as same itself divides ?
Are there two which choose to mingle,
So that one each other hides ?

As the answer to such question
I've found a sense that's true:
Is it not my songs's suggestion
That I'm one and also two ?

Johann Wolfgang Goethe (1749-1832)

Translation of Goethe's poem taken from [Webpage Weimar]; Photo of Ginkgo leaf taken from [Webpage Schwabe].

1 Introduction

1.1 Motivation

Projects set up to design, implement, and deploy software for supporting cooperative work (e.g. group calendars, knowledge management systems, group editors) have a bipartite character: if they are successful, they yield a so-called „socio-technical[1] system" in which the technical system software is matched by a social system that includes work procedures making expedient use of the software. The challenge for the field of **c**omputer **s**upported **c**ooperative **w**ork (CSCW) is that these projects need to integrate processes of software engineering and processes of organizational change in order to allow a new „socio-technical system" to emerge; and that these processes are of different nature and require different methodological support for their successful evolvement.

The goals of this thesis are the provision of theoretical concepts describing „socio-technical systems" and the elaboration of methodological support for their successful evolvement in the course of CSCW-projects.

1.1.1 Problems with Software supporting Cooperative Work

There are many reports in scientific as well as in popular publications describing how the interplay between software and organizational procedures does not fulfill the expectations:

- Orlikowski [(1992a)] analyzes the deployment of Lotus Notes® in an American company and identifies several "organizational issues" preventing Lotus Notes® from being used as originally planned by the management. One of the problems she lists is the lack of procedures and policies concerning e.g. data security which unsettled potential users. Another problem was that management was not successful in making Lotus Notes® known as a groupware and that the system was consequently used for supporting individual work rather than for cooperative work processes.

[1] I follow Cummings and Srivastva and many of the early researchers from the Tavistock Institute in using the hyphenated form „socio-technical" rather than „sociotechnical", because „*The hyphen between the words socio and technical represents the directive correlation or matching that must occur between the independent but correlative systems to reach a desired outcome.*" [Cummings, Srivastva (1977), p. 58].

- Grudin [(1988)] reports the non-use and even sabotage of several types of groupware systems. One of his examples refers to a project management system. This system offered the possibility of indicating problems occurring in the context of project management. A user took advantage of this feature and indicated a problem in finding task priorities. The CSCW-system reacted by automatically requesting an increased number of progress reports from this user; the user who was surprised by this reaction was also confronted with the request to forward the progress reports directly to his management. As a result the users avoided indicating problems within the system. Some even wrote programs for automatically opening files and changing their dates such that the automated monitor would not notice that they were not using the system anymore [ibid. p. 89].
- In a popular German computer magazine [Birkelbach, Schulzki-Haddouti (2004)], it is described how the organizational structures of a civil authority do not match the increased usage of e-mail and that users print out e-mails for giving them into internal mail and further processing by the central post-office.
- An article in another computer magazine discusses so-called virtual teams [Computer Zeitung Nr. 9 / February 24th 2003] and elaborates the fact that spatially distributed teams relying on IT-systems for their coordination and communication need explicit rules to augment the technical system for their organization of work.

Any organization investing in the design and implementation of a software system to support cooperative work (CSCW-system) does so because it has certain expectations concerning the usage of the system. These expectations may be related to faster exchange of information, better coordination among teams, shortened process durations etc.; in any case the organization expects a return on investment that can only be reached if the new software system is not only used, but also used in a predictable way. The examples above illustrate how organizational procedures and software features have to be coordinated for successful use.

1.1.2 Limitations Regarding the Design of Work Procedures

The discipline of software engineering has reacted to the growing importance of organizational procedures complementing the software system by including additional tasks into the design process. Sommerville [(2004), p. 37] argues that operational procedures are needed for the successful deployment of software and that *"these operational processes have to be defined and documented during the system development process"*. The Rational Unified Process (RUP) includes a discipline dedicated to *„the type of activities that take place when the software product is deployed in its operational environment and of the additional artifacts that may need to be developed for its successful use."* [Kruchten (2004), p. 237]. These suggestions provide good starting points for integrating organizational change which leads to new work procedures making use of a CSCW-system. However, it may not be assumed that work procedures ensuring the "successful" usage of a software system can be designed and put into operation just as the software itself can be designed and implemented. Research in the field of CSCW has shown that the deployment of software in an organization does not directly and deterministically lead to corresponding work procedures. Instead complex processes take place during which the organization changes to incorporate the new CSCW-system into its procedures. Orlikowski was the first to systematically document and analyze the way in which an organization reacts to new groupware, and as a result she conceived the concept of „metamorphosis" [Orlikowski (1996)]. "Metamorphoses of groupware usage" describe how users adopted a new groupware in their work procedures and how they iteratively came to new ways of using it, which were not planned in the original design. Later, the term „evolving use" [Andriessen, Hettinga, Wulf (2003)] was used for describing the phenomenon that

> *„More often than not, users appear to use groupware in another way than the groupware designers intended or IT-departments expected. Users tend to 're-invent' the technology by developing novel uses, giving rise to what we call 'the evolving use of technology'."*

A third term which is now used in this context is the term „appropriation". Appropriation denotes the social processes by which an organization deals with a new CSCW-system ([DeSanctis, Poole (1994)]; [Dourish (2003)]; [Pipek (2005)]); this may include organizational changes as well as changes to the software in form of configuration or end-user development.

There is an inherent conflict between the need of ensuring a certain usage of a software system by "designing" corresponding operational procedures and the observation that appropriation is a self-contained process which is hardly controllable. However, a software engineering project, set up to design, implement and deploy a CSCW-system, cannot be contented with the insight that the question whether and how the software might be used, will only be resolved in a process of appropriation after its deployment. Methods supporting planned organizational change in the context of the design, implementation and deployment of a CSCW-system are needed. But such methods have not yet been in the focus of interest within the field of CSCW. About 10 years ago Kuutti diagnosed [(1996), p. 189; bold emphasis added]:

> *„It [the CSCW community] has neither been nor is it now oriented toward* ***organizational change****. If we divide the community coarsely into two groups – „technical tinkerers" and „social scientists" – the former are interested in the implementation of their own system and the latter in how people use or misuse systems, are disturbed by them in their work or are able to circumvent their problems. Both groups are largely lacking a perspective and interest in design in general and* ***in changing work processes or organizations by using systems****."*

This situation has not changed significantly (cf. Section 4.3 for an overview and discussion of methods in the field of CSCW), and this thesis intends to contribute to closing this gap.

1.1.3 Socio-Technical Self-descriptions (StSd)

The problem for CSCW-projects can be recapitulated as follows:

1. CSCW-projects are successful only if they yield a „socio-technical system" in which software features match with corresponding work procedures.
2. The design of work procedures in analogy to the design of software as a technical system does not necessarily lead to a „socio-technical system", because the adoption of software in an organization is a social rather than a technical process.

3. Methods are needed to support an organization in planning and establishing work procedures that make use of a new CSCW-system.

I suggest turning to Luhmann's concepts of newer system theory which first allow the description of self-contained processes such as appropriation of CSCW-systems in organizations; and which then support the elaboration of methods for intervening into such processes.

Within newer system theory organizations are viewed as operationally closed systems which react to changes in their environment according to their own internal rules. Consequently organizations can be triggered to change, but the change itself cannot be determined from the environment. The concept of self-description is important for explaining how social systems such as organizations maintain an identity although they do not possess a physical boundary. In a very intuitive sense, self-descriptions describe the main characteristics of an organization and thereby define the expectations that are directed toward its members. Examples of self-descriptions include structural charts showing the affiliation of members to sections, mission statements that describe the organization's purpose and goal, values and norms that are known and valid within an organization. Self-descriptions fulfill an interesting dual purpose: First, they stabilize the organization by ensuring its identity. Second, they enable processes of change by offering something to reflect, question and change.

I transfer the concept of self-description into a concept of socio-technical self-description (StSd) for supporting CSCW-projects. Again, in a very intuitive sense, socio-technical self-descriptions allow an organization to describe its own computer supported work and agree to conventions necessary for coordinating the use of the CSCW-system. Simple examples for socio-technical self-descriptions are:

- the convention to store the CVs of all team members in a certain directory of a knowledge-management system;
- the agreement that entries in the team's group calendar should be updated each Thursday;
- the decision to make a team member responsible for personally reminding the others to read new documents in a document management system;
- the information in ISO9000 certification documents that SAP® is used for financial accounting.

By supporting an organization in creating socio-technical self-descriptions in the course of a CSCW-project, the social process of appropriation can be initiated in parallel with the process of software engineering. From this I expect the effect that a better match between software features and organizational procedures is achieved which should then lead to a more successful usage of the software system. The task for this thesis is to take this basic idea and derive concrete methodological support for CSCW-projects. The question that I strive to answer is: **How can the concept of self-description from newer systems theory be used for improving the co-evolvement of software engineering and organizational change in CSCW-projects?**

1.2 Scope

1.2.1 Software Engineering in Context of CSCW-Projects

It has been mentioned several times now, that this thesis is concerned with a special type of software, namely with CSCW-applications. A brief introduction into the contents and termini of CSCW shall be inserted at this point.

Computer Supported Cooperative Work (CSCW) became relevant as a research field within computer science in the late 1980's. Schmidt and Bannon [(1992)] provide a summary of the development from which they derive a definition that is widely accepted. To allow a clear distinction between the research field and the cooperative work itself, I will use the term „field of CSCW" as follows:

The term **field of CSCW** shall be used within this thesis whenever the field of research is addressed that has been defined by [Schmidt, Bannon (1992), p. 11] as follows:

> *"CSCW should be conceived of as an endeavor to understand the nature and requirements of cooperative work with the objective of designing computer-based technologies for cooperative work arrangements."*

Term 1: Field of CSCW

The term CSCW alone shall then be used to refer to the work processes themselves

The term **computer supported cooperative work** and its acronym **CSCW** shall be used within this thesis to denote processes of cooperative work which make use of IT-systems for supporting the cooperation.

Term 2: Computer Supported Cooperative Work (CSCW)

This definition implies that the IT-systems are not only available but actually used. There are other definitions that only require the availability of IT-systems, not their use [Oberquelle (1991), p. 5]. There might be settings in which IT-support is available but not used; and it might be interesting to analyze the conditions of such settings, but too many settings would fall under the category of CSCW if only the existence and not the usage of IT-systems were required in the definition. Therefore I use the term CSCW in the narrow sense, requiring that computer support is actually used.

The fact that CSCW-systems aim at supporting people in cooperative work, distinguishes them from software that is intended to support single users. Phenomena in the context of use of CSCW-systems are not limited to issues concerning individuals (e.g. ergonomics); they are also located on the social level of people interacting with each other. One definition of cooperative work which also addresses some issues relevant for designing CSCW-systems is given by [Schmidt, Simone (1996), p.158] and I adopt it for this thesis.

Within this thesis, **cooperative work** shall be understood as suggested in the following definition by Schmidt and Simone [(1996), p. 158]:

> *„Cooperative Work is constituted by the interdependence of multiple actors who, in their individual activities, in changing the state of their individual field of work, also change the state of the field of work of others and who thus interact through changing the state of a common field of work."*

Term 3: Cooperative Work

The design, implementation and deployment of software systems are usually organized in form of projects. Projects – unlike processes – are characterized by being planned, having a goal and a schedule[2]. This thesis regards projects in the context of CSCW. The term „CSCW-project" shall be used and a preliminary definition (which will be extended chapter 3 in Term 23) is:

A **CSCW-project** is a project that has the goal of designing, implementing and deploying a CSCW-system in an organization. A successful CSCW-project results in a technical CSCW-system and corresponding work procedures within the organization.

Term 4: CSCW-Project (preliminary)

The phenomena metamorphosis, evolving use and appropriation described above are processes and no projects. I will show that the methodological support elaborated in this thesis reaches into these processes, but the emphasis of this thesis is put on CSCW-projects. My assumption is that if the appropriation of a CSCW-system is methodologically supported and initiated during a CSCW-project, it will lead into a continuous and conscious process of appropriation accompanying the use of the software.

[2] The Merriam-Webster Dictionary defines project as "a planned undertaking". Cp. "project." Webster's Third New International Dictionary, Unabridged. Merriam-Webster, 2002. http://unabridged.merriam-webster.com (29 Aug. 2005).

1.2.2 Transdisciplinary Character of this Thesis

The evolvement of a „socio-technical system" that integrates a CSCW-system and corresponding work procedures is a transdisciplinary endeavor in the sense Mittelstraß coins the word [Mittelstraß (1998), pp. 45, translated from German]:

> *Transdisciplinary research fulfills transdisciplinary expectations of the real world („Lebenswelt") where disciplinary knowledge alone is not sufficient to solve our problems.*

The „Lebenswelt" of CSCW-systems requires the integrated evolvement of software and corresponding work procedures and neither computer science nor social science alone is able to provide adequate theoretical insights and methods.

While the discipline of software engineering provides theoretical frameworks as well as practical methods for the definition and production of software-systems; it neither contains methods to describe the impact a software-system has within an organization nor methods to actively support the social processes necessary to develop work processes that match the software-system. This thesis is transdisciplinary because it examines how theories and methods stemming from social sciences can be added to methods of software engineering such that CSCW-projects have an integrated set of methods at their disposal for fulfilling their task of creating a socio-technical system.

The social theory to augment software engineering within this thesis is newer system theory. The next section motivates this choice.

1.3 Social Theories discussed in the Field of CSCW

1.3.1 Computer Science and Theories from other Disciplines

There are many research areas in which computer science turns to theories stemming from other disciplines for supporting the elaboration of methods and systems: Concepts from social sciences that describe and explain social behavior are used for designing independent agents in multiagent systems [Wooldridge (2002), p. 11]. Game theory [Dixit, Skeath (2004)] is becoming increasingly interesting for computer science: one purpose is to use it for the design of multiagent systems [Wooldridge (2002)]. Another application is the description and prediction of users' behavior in decentralized settings like the internet. Speech act theory – as one example of communication theories – has been used for designing workflow systems [Winograd (1988)] as well as agent communication languages [Wooldridge (2002), pp. 164]. Models from cognitive psychology are used for deriving methods of usability testing [Cockton, Lavery, Woolrych (2003)].

Due to the closeness of technical and social issues it is not surprising that within the field of CSCW there is a tradition of discussing social theories and evaluating their potential for explaining phenomena occurring in the context of cooperative work. The theories discussed most are: activity theory, structuration theory, actor-network theory and systems theory. The following sections provide a brief overview of each theory and its usage within the field of CSCW.

1.3.2 Activity Theory

Activity theory is a social theory that provides means for explaining individual human actions in a social context. It became interesting for CSCW because it overcomes the limitations of cognitive theories used in the HCI context by including social phenomena and their influence on individual actions. While cognitive theories explain individual behavior and can support the design of man-machine interfaces they fall short in explaining social phenomena of cooperative work. Thus they are of limited use in informing the design of IT-systems meant to support cooperative work.

Activity theory is linked to the names of the Russian psychologists Vygotsky, Leont'ev and Luria (for an introductory overview see [Engeström, Miettinen (1999)]). In their quest to explain the relation between the individual action and societal circumstances, the differentiation between action and activity was made. The term action refers to something an individual does while activity denotes the broader social context which includes several actors. A famous example is hunting as a collective activity by a group of men within which each individual performs different actions.

Activity theory views human action as being mediated by objects such as tools that carry in them the cultural history of mankind. For answering the puzzling question how human action is conditioned by social structures which are themselves brought forth by human actions, activity theory describes two processes which are „*inseparably intertwined*": internalization and externalization [ibid. p. 10]. Internalization is the process by which culture determines human action and ensures continuity. Externalization is the process by which human actions construct *„new instruments and forms of activity at collective and individual levels"* and thereby initiate social change [ibid. p. 11]. The interplay between externalization and internalization can be used to explain processes within an organization that „externalizes" some of its social rules into a CSCW-system and thereby forces its members to „internalize" these rules again by using the CSCW-system.

But there is another reason why researchers use activity theory in the context of CSCW. CSCW was (and in fact still is) in search of a theory that describes cooperative work such that it can be used to analyze workplace settings for the design of appropriate CSCW-systems. Engeström's [ibid. p. 9] reasoning explains why activity theory is one option for such a theory:

> *„To be able to analyze such complex interactions and relationships, a theoretical account of the constitutive elements of the system under investigation is needed. In other words, there is a demand for a new unit of analysis. Activity theory has a strong candidate for such a unit of analysis in the concept of object-oriented, collective, and culturally mediated human activity, or activity system. Minimum elements of this system include the object, subject, mediating artifacts[3] (signs and tools), rules, community, and division of labor … ."*

[3] In German the word "artifact" is often connotated with "an unintended artificial effect". This is not the meaning in which the word is used in this thesis. In English and especially within the terminology of CSCW-related research, the word "artifact" refers to something that is man made.

The Merriam-Webster Dictionary defines artifact as: "a usually simple object (as a tool or ornament) showing human workmanship or modification as distinguished from a natural object" ("artifact." Webster's Third New International Dictionary, Unabridged. Merriam-Webster, 2002. http://unabridged.merriam-webster.com (29 Aug. 2005))

CSCW-research needs to understand the nature of technically supported cooperative work. Activity theory provides means for describing and explaining cooperative action and is thereby able to support research interests within CSCW. Activity theory is used today to interpret findings of ethnographic work and to draw conclusions for the design of CSCW-systems (e.g. [Luff, Hindmarsh, Heath (eds.) (2000)], esp. [Engeström (2000)]).

1.3.3 Structuration Theory

In his book *„Constitution of society"* [Giddens (1984)] Giddens lays out a sociological theory that attributes the existence of social macro-level structures such as states to human action. He prefers the term „structuring" to „structure" because it emphasizes the dynamic qualities of social structures: although they limit and influence human activities they are at the same time the result of human activity and subject to change. Giddens uses the term *„duality of structure"* in this context (cp. [Joas, Knöbl (2004), pp. 393]).

Although Giddens himself never applied his structuration theory to information system development, there is a long tradition of doing so starting in the mid-1980's [Jones (1999)]. Wanda Orlikowski elaborated her much observed concept of „duality of technology" [Orlikowski (1992b)] on Giddens' structuration theory; and DeSanctis and Poole used it as one source for their Adaptive Structuration Theory (AST) [DeSanctis, Poole (1994)]. They were interested in the finding out why sometimes the deployment of an IT-System leads to the desired organizational change and sometimes not. *„We propose adaptive structuration theory (AST) as a framework for studying variations in organization change that occur as advanced technologies are used"* [ibid. p. 122]. Poole and DeSanctis analyzed how structures given by IT-systems (here especially group decision support systems) influence human action and subsequently the emergence of new social structures. The latter are indicators of organizational change induced by the IT-system. The unit of their analysis is *„a group's application of a specific technology-based rule or resource within a specific context and at a specific point in time"* [ibid. p. 128]. With reference to Ollman, they define the term „appropriation" as *„the immediate, visible actions that evidence deeper structuration processes"*. The idea of appropriation has also influenced the way CSCW-researchers conceptualize the usage of CSCW-system in organizations and draw conclusions for their design (e.g. [Dourish (2001b), p. 171], [Dourish (2003)]).

Another employment of Giddens' structuration theory within CSCW is for the explanation of the interdependencies between methods for the development of information systems and the determining organizational factors. The choice of specific design methods depend on the organizational context; but once it has been chosen and put into practice, any design method will influence the organizational context. Structuration theory is used as the theoretical framework of the methodology Multiview II in this sense [Avison et al. [1998)].

1.3.4 Actor-network Theory

Actor-network theory goes back to the French sociologist Bruno Latour. He puts human action into a network of agents by which it is influenced. In these actor-networks he deliberately does not distinct between human and non-human agents. His point of view is that human action is determined by the agents with which it is linked; whether these are of human or technical nature does not alter their influence. Furthermore Latour views technical artifacts as containing and proliferating cultural agreements [Latour (1998), p. 81]. One of his famous examples are the thick knobs attached to hotel keys which prevent guests from taking the key outside in their pockets and forgetting them there. He assumes that early inn-keepers grew tired of asking their guests not to take the key outside; after a while they conceived a device which inherently carried their request. Latour calls the process of transferring cultural agreements into technical artifacts inscription.

Latour's theory is able to put the growing importance of technical artifacts into relation with human actions; his approach takes care of the social component inherent in man-made artifacts (which provides a link between ANT and Activity Theory). Within the field of computer science, ANT is less popular than activity theory or structuration theory but some authors such as [Monteiro, Hanseth (1996)] argue that it is more appropriate to describe in detail how IT-systems are socially constructed. Following this line, Klischewski uses ANT for conceptualizing *"systems development as networking"* [Klischewski (2002)]; he describes how the successful development and use of an information system in large organizations can be described as the creation of a network of actors and commitments made by these actors.

1.3.5 System Theory

In 1984 (the same year in which Giddens published his main work on Structuration Theory) Luhmann published his main work on the theory of social systems (German: [Luhmann (1984)], English: [Luhmann (1995)]). In contrast to Giddens Luhmann's theory is based on the premise that social phenomena (e.g. organizations, states, societies) cannot be explained by human actions alone but possess new qualities of their own and follow their own logic. Luhmann's approach was first based on traditional systems theory but then he performed a paradigmatic change by looking at the mechanisms that reproduce social systems rather than the elements from which they are made.

Newer systems theory as elaborated by [Luhmann (1995)] views organizations as social systems that are operationally closed. Being operationally closed means that the sequence of basic operations that constitute the system is determined exclusively from within the system. According to newer system theory organizations consist of a network of communications that continuously brings forth new communications which relate to previous ones and prepare the ground for future communications. From the outside (e.g. by a consultant or a new CSCW-system) this network of communications can be irritated (perturbed) but never determined.

For his theory of social systems Luhmann draws on concepts elaborated by the Chilean biologists Maturana and Varela (cf. section 2.3.2). In 1986 Winograd and Flores published their book on *„Understanding Computers and Cognition"* [Winograd, Flores (1986)] where they use Maturana's and Varela's insights on human cognition to argue for a different perspective in designing computer based systems. They state that the influence a computer system has on the flow of *„conversations"* within an organization is more important than its objective features [ibid, p. 173]. Although Winograd and Flores do not relate their work to Luhmann, they share the same conceptional roots, which have thus been employed in the realm of computer science before.

Within the field of CSCW newer systems theory has been used explicitly to explain phenomena of organizational change following the deployment of a groupware system ([Wulf (1999)], [Andriessen, Hettinga, Wulf (2003)]). It proved especially useful in explaining why the way an organization adapts to new software cannot be predicted. Loser used Luhmann's system theory to explain how organizations adopt commercial off the shelf software, and to derive a method for supporting processes of adoption [Loser (2005)].

There is yet another branch of systemic research that had an impact on CSCW-research: Checkland's widely received Soft Systems Methodology (SSM, cf. section 4.3.3, p.118). SSM uses the concept of *„appreciative systems"* [Checkland (1999), pp. A50] that also incorporates Maturana's and Varela's concepts [ibid, p. 52] and characterizes social systems as being determined by their own history rather than by interventions from the outside.

1.3.6 Selection of Systems Theory for this Thesis

From these theories system theory is the one that has been selected as the foundation for this thesis. There are a number of reasons why I think it worthwhile to investigate in detail whether and how the insights of system theory can add to the methods of software engineering in a CSCW-project:

1. In [Luhmann (1995)] system theory sets out by describing systems on an abstract level that includes explanations of technical as well as social systems. From there on Luhmann discusses the special properties of social systems including their relation to other types of systems. Thereby, systems theory offers a theoretical framework to elaborate the interplay between technical and social systems within the context of CSCW.
2. The term „system" is ubiquitous in computer science. Not only is it used to denote technical devices but also to conceptualize the interdependencies between people, information and computers in so-called information systems. This indicates that there are anchors within the terminology of computer science to which system theory can be linked. But system theory also provides a theoretical framework to indicate problems that are inherent to some concepts of „systems" used in computer science.
3. Systemic concepts have been used within CSCW-research before; therefore there are publications, methods and discussions to which the findings in this thesis can be connected. The methodological consequences of systemic insights in the context of software engineering however have not yet been fully explored.
4. The insights of newer systems theory have been used to develop methods for intervention into organizations also referred to as „systemic intervention". These methods aim at purposefully changing an organization without ignoring the fact that social processes cannot be engineered in the way software can. So in addition to an elaborated theoretical framework system theory provides a set

of proven methods which can be analyzed with respect to their usefulness in the context of CSCW projects.

The social theories listed above are in part competing with each other and in part complementing one another. It is beyond the scope of this thesis to analyze and evaluate the explanatory potential of the different theories. System theory is the base for the methods I elaborate; therefore its concepts (and not the others') are discussed in detail. But references to the other theories are provided where appropriate.

1.4 Approach, Structure and Results

The answer to the question how the concept of self-description from newer systems theory can be used for improving the co-evolvement of software engineering and organizational change in CSCW-projects is given in four steps:

- First, I elaborate a theoretical foundation which includes the concepts of socio-technical organization, socio-technical setting and socio-technical self-description (chapter 3).
- Second, I elaborate an initial methodological concept supporting the creation and maintenance of socio-technical self-descriptions in CSCW-projects (chapter 5).
- Third, I analyze empirical evidence from two explorative case studies for improving the initial methodological concept (chapter 7).
- Fourth, I derive a finalized methodological concept in form of recommendations how socio-technical self-descriptions should be used in CSCW-projects (chapter 8).

Chapter 2 lays the theoretical foundation for my research by discussing relevant aspects of system theory: Section 2.1 provides an overview of existing conceptualizations of “socio-technical systems”, discusses their advantages and disadvantages and derives the motivation for a reconceptualization. Section 2.2 defines the “technical side” of socio-technical systems with respect to CSCW-projects. Section 2.3 introduces relevant aspects of Luhmann's theory of social systems. Besides explaining Luhmann's concepts, I continuously discuss the applicability of this social theory for the field of CSCW. In section 2.4 I gather Luhmann's dispersed explanations regarding the concept of self-description and derive four characteristics of self-descriptions which lay the base for using the support of self-descriptions as part of my methodological concept.

In chapter 3 I extend Luhmann's concepts by explicitly including CSCW-systems into the terms and models; here the concepts of „socio-technical organization” and „socio-technical self-description” are introduced. In order to place my theoretical concepts within the field of CSCW, they are compared to established concepts such as articulation work, appropriation, groupware conventions and boundary objects. At the end of chapter 3 the theoretical base is laid for integrating software engineering and processes of organizational change in CSCW-projects by supporting the creation of socio-technical self-descriptions; I discuss methodological implications for CSCW-projects and derive an initial process model for CSCW-projects.

Chapter 4 analyses different approaches for integrating software engineering and organizational change; the goal of chapter 4 is to find out in how far existing methods and approaches could support the creation and maintenance of socio-technical self-descriptions. Process models and methods from software engineering (section 4.1), systemic intervention (section 4.2) and design methods developed in the field of CSCW (section 4.3) are discussed in the light of my theoretical concepts. A summative discussion (section 4.4) identifies best practices as well as methodological gaps.

Based on the theoretical concepts of chapter 3 and the best practices derived from the state of the art in Chapter 4, Chapter 5 introduces my initial methodological concept of working with socio-technical self-descriptions in CSCW-projects. Within this initial methodological concept I describe socio-technical self-descriptions in terms of their content and their form, I introduce the workshop design socio-technical walkthrough (STWT) as a communicative setting for creating and maintaining socio-technical self-descriptions, and I describe how socio-technical self-description can link the two processes of software engineering and organizational change by serving as boundary objects.

I conducted two case studies for exploring and evaluating my initial methodological concepts. Chapter 6 describes the settings of both case studies. Case Study 1 is called “Mobile Communication System” and deals with the deployment of mobile information technology for improving the communication between drivers and dispatchers in a company offering logistics services. The CSCW-system designed and evaluated is called “SpiW-Com” (cf. section 6.2). Case Study 2 is called “Electronic ToC-Service” and is about the replacement of a paper-based procedure for ordering articles from scientific periodicals by an electronically supported procedure. The setting is a group of about 8 researchers at a German university and the name of the CSCW-system designed and deployed is “ELISE” (cf. section 6.3).

In chapter 7 I discuss my initial methodological concept in the light of the empirical evidence gained from the case studies. Based on a set of 10 research questions, I provide a detailed analysis of the case studies leading to further insights concerning the contents and form of socio-technical self-descriptions, the support of their creation and maintenance by means of facilitated workshops and their ability to integrate organizational change and software engineering by serving as boundary objects.

In chapter 8 I consolidate the theoretical concepts and empirical insights to provide a framework for using socio-technical self-descriptions within CSCW-projects (section 8.1). My framework consists of three parts:

1. The theoretical concepts describing interdependencies between an organization and a CSCW-system: socio-technical setting, socio-technical organization and socio-technical self-description
2. A process model for CSCW-projects that integrates the processes of organizational change and technical software engineering
3. Methodological support for documenting socio-technical self-descriptions in CSCW-projects:
    a. Regarding their contents

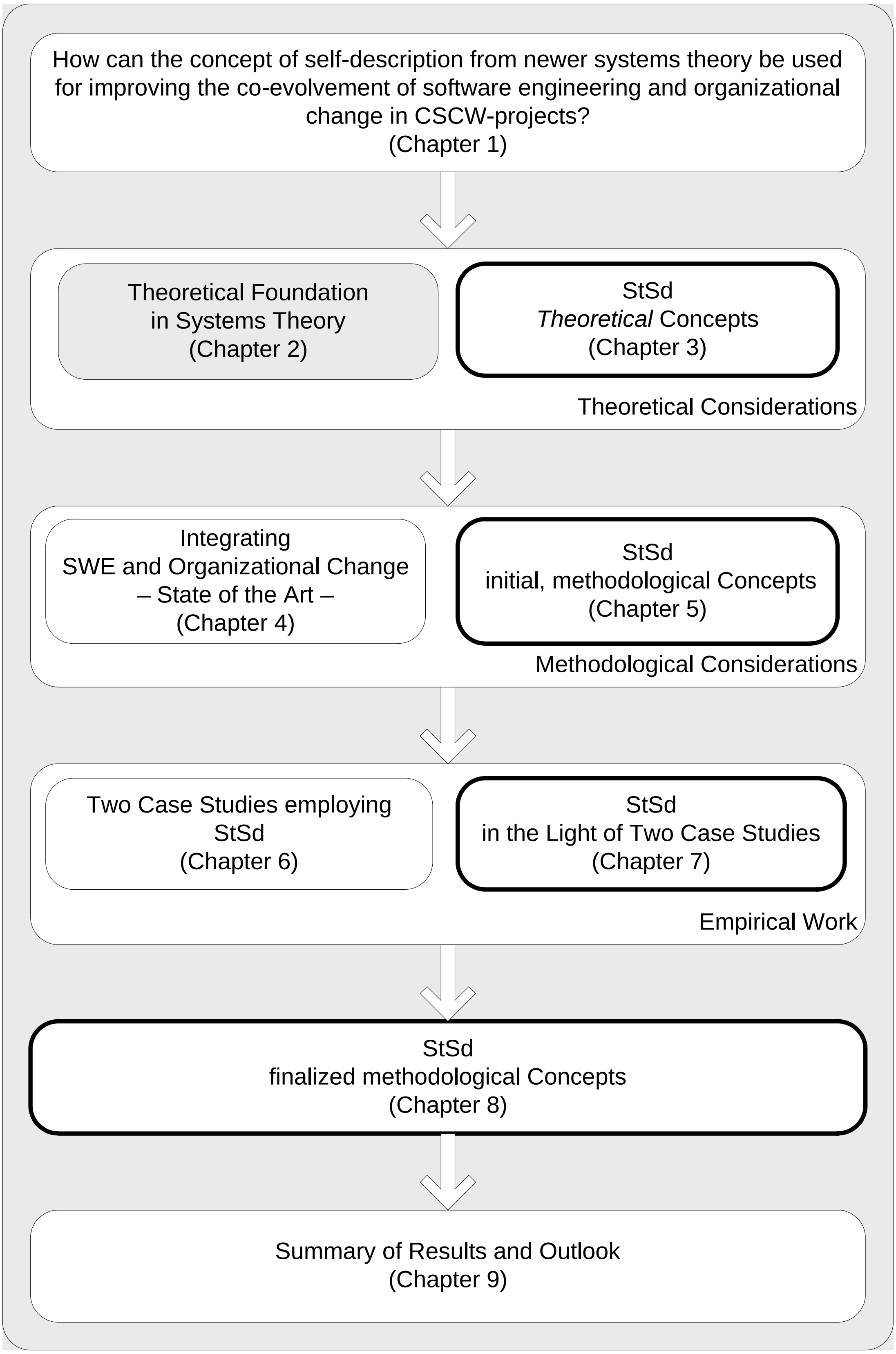
b. Regarding their form

c. Workshop design socio-technical walkthrough

Socio-technical self-descriptions are meant to be but one methodological component within CSCW-projects. Therefore I present the methodological support in form of recommendations that can be employed in combination with other methods supporting CSCW-projects (section 8.2). For validating that it is indeed feasible to combine socio-technical self-descriptions with other process models or techniques, I discuss two examples in detail: the integration of StSd with a process model prominent in the field of software engineering – the Rational Unified Process (cf. section 8.3.1); and the integration of StSd with a method prominent in the field of CSCW – MUST (cf. section 8.3.2). Further options for integration are discussed briefly in section 8.3.3.

Chapter 9 concludes by giving a summary of the main results and providing an outlook to future work.

All pieces of empirical data (diagrams, transcripts from interviews, questionnaires etc.) directly needed for the reasoning in this thesis are included in English language within the text. A supplementary appendix is available in Chapter 11; it contains a larger set of empirical data in original, German, form.



How can the concept of self-description from newer systems theory be used for improving the co-evolvement of software engineering and organizational change in CSCW-projects?
(Chapter 1)

Theoretical Foundation
in Systems Theory
(Chapter 2)

StSd
Theoretical Concepts
(Chapter 3)

Theoretical Considerations

Integrating
SWE and Organizational Change
– State of the Art –
(Chapter 4)

StSd
initial, methodological Concepts
(Chapter 5)

Methodological Considerations

Two Case Studies employing
StSd
(Chapter 6)

StSd
in the Light of Two Case Studies
(Chapter 7)

Empirical Work

StSd
finalized methodological Concepts
(Chapter 8)

Summary of Results and Outlook
(Chapter 9)

2 Theoretical Foundation in Systems Theory

Chapter 2 lays the theoretical foundation for my research by discussing relevant aspects of system theory: Section 2.1 provides an overview of existing conceptualizations of "socio-technical systems", discusses their advantages and disadvantages and derives the motivation for a reconceptualization. Section 2.2 defines the "technical side" of socio-technical systems with respect to CSCW-projects. Section 2.3 introduces relevant aspects of Luhmann's theory of social systems. Besides explaining Luhmann's concepts, I continuously discuss the applicability of this social theory for the field of CSCW. In section 2.4 I gather Luhmann's dispersed explanations regarding the concept of self-description and derive four characteristics of self-descriptions which lay the base for including the support of self-descriptions as part of my methodological concept.

2.1 What are "Socio-technical Systems"?

2.1.1 The Need for a Definition

It has been argued in section 1.1 that the outcome of a CSCW-project is twofold: a technical CSCW-system matched by corresponding work procedures in the organization. The interdependencies of technical and organizational aspects inspired the term „socio-technical" system to describe an entity in which organizational and technical subentities are integrated. The advantage of having a construct like „socio-technical system" is that the goal of a successful CSCW-project can be described, and methods for achieving the goal as well as indicators measuring their success can be derived. To serve these purposes, a concept of „socio-technical system" for CSCW-projects should

1. Allow the description of the outcome of a CSCW-project.
2. Be suitable for deriving methodological support for CSCW-projects.
3. Allow the description and explanation of known phenomena during the appropriation of CSCW-systems in organizations.

The term "socio-technical system" is often used in a very intuitive sense whenever the closeness and the interdependencies between technological and social aspects are to be emphasized. Existing conceptualizations or mere usages of the term are discussed in the following sections to derive the motivation for a reconceptualization.

The Merriam-Webster dictionary defines a system as *„a complex unity formed of many often diverse parts subject to a common plan or serving a common purpose"*[4]. This wide definition is a common basis for a number of ways in which the term „socio-technical system" is used. The following overview of definitions starts some 50 years ago.

2.1.2 Tavistock's Socio-technical Systems

To expose the origins of the term socio-technical systems one has to turn to projects done by researchers of the Tavistock Institute of Human Relations in British coalmines in the early 1950s. Traditionally the task of coal-getting was accomplished by small groups of men that took shared responsibility for a mining site; each of the men *„experiences the entire cycle of operations within the compass of its membership."* [Trist, Bamforth (1951), p. 6]. In the postwar era, the deployment of the technically innovative longwall method was accompanied by new ways of organizing coal mining: men worked in three shifts, each responsible only for parts of the process [Trist, Bamforth (1951), p. 4]; the principle "*one man – one job*" [Trist et al. (1963), p.13] often replaced the group oriented work organization which allowed a less rigid division of labor.

The technological innovation however did not lead to an improvement of the bad situation of Great Britain's mining industry, and research was initiated to analyze causes and find solutions [Trist, Bamforth (1951), p. 5]:

> *„Faced with low productivity despite improved equipment, and with drift from the pits despite both higher wages and better amenities, those in authority have increased their interest in the organizational innovations that have been taking place. … the longwall method will be regarded as a technological system expressive of the prevailing outlook of mass-production engineering and as a social structure consisting of the occupational roles that have been institutionalized in its use."*

Besides analyzing the situation in coal mines in which the new longwall method lead to new working arrangements, the researchers also found places where the traditional organization of semi-autonomous groups was combined with new technology [Trist (1993), pp. 37].

These projects (followed by more projects in different industries and countries) inspired the so-called "socio-technical approach". The researchers came to the conclusion that *"Considering enterprises as 'open socio-technical systems' helps to provide a more realistic picture of how they are both influenced by and able to act back on their environment"* [Emery, Trist (1960), p. 94]. This statement reflects the researchers' occupation with system theory of their time. Rather than viewing organizations as closed systems which are determined by their internal structure, they took up new ideas introduced e.g. by Ludwig von Bertalanffy [ibid. p. 84] and described organizations as open systems which are determined by the interchange with their environment.

Although the notion of open systems was much discussed and reflected ([Emery, Trist (1960)], [Cummings, Srivastva (1977), pp. 49]), clear definitions of what a socio-

[4] Webster's Third New International Dictionary, Unabridged. Merriam-Webster, 2002. http://unabridged.merriam-webster.com (11 Sep. 2004)

technical system should be are hard to find [Sydow (1985), p. 27]. One definition is [Cummings, Srivastva (1977), p. 60]:

> *"A socio-technical system is a nonrandom distribution of social and technological components that coact in physical space-time for a specified purpose."*

The following description might capture the theoretical ideas and their practical impact even better [Cummings, Srivastva (1977), p. 60]:

> *"Socio-technical systems theory is based on two fundamental constructs: directive correlation and open system theory. The former provides the basis for conceiving of work systems in which human beings interact with technological components as being composed of two independent but correlative systems – a social system and a technological system – while the latter furnishes a basis for viewing such systems as being interdependent with their environment and hence, open socio-technical systems."*

With respect to CSCW-projects, these definitions fulfill the purpose of describing the fact that the technical CSCW-system and the organization using it are interdependent and that they form a larger unit named "socio-technical system". Especially the notion of *"independent but correlative"* will be taken up again in my reconceptualization of socio-technical systems. However, the early definitions and descriptions are very vague with respect to the actual interaction between the social and technical component.

Since the first definitions of „socio-technical system" were elaborated some 50 years ago, they obviously included no link to IT-systems as they are designed, implemented and used today. There are more recent conceptualizations of the term „socio-technical system" which explicitly relate to IT-systems. The next sections discuss some typical examples.

2.1.3 „Socio-Technical Systems" in IT-Design

Information Systems as Socio-technical Systems

For the field of information management Krcmar describes information systems as socio-technical systems [Krcmar (2005), p. 25, translated from German]:

> *Information systems are sociotechnical (man-machine-) systems that comprise human and technical components (subsystems). Information systems are employed with the goal to economically optimize the provision of information and communication.*

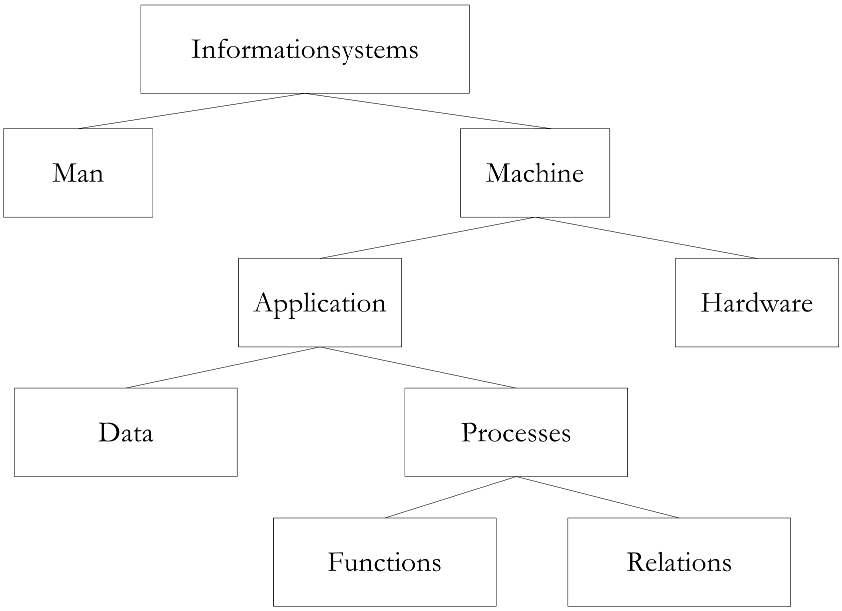


Figure 1: Information system as Man-Machine-Systems
(Translated from [Krcmar (2005), p. 25])

This definition acknowledges that human as well as technical aspects need to be regarded when *„employing"* information systems (similar usages of the concept "socio-technical system" can be found in [Heinrich (2002), p. 1041], [Sinz (2002), p. 1070]). For a number of reasons however this definition is not suitable for describing processes occurring in the context of a CSCW-project.

First, the „social" side in Figure 1 is reduced to *Man* („Mensch") only. While this might be suitable for describing aspects of man-machine interfaces, it does not support the description of the social phenomena determining cooperative work. Second, the phrase that *„socio-technical systems are employed"* implies that the human components can be *„employed"* just as the technical can; this does not support the explanation of phenomena like evolving use.

Galliers and Swan include an organizational perspective into the definition of information systems by saying that these are *„not just IT-based systems but socio-technical systems which might incorporate IT, work processes, routines, and wider organizational arrangements such as training programmes"* [Galliers, Swan (2000), p. 81]. In this definition work processes are explicitly mentioned to be part of a „socio-technical system". However the authors are quite vague about the components of their „socio-technical system" and they do not provide any information about the way in which the different components are related to each other.

Software as a Part of Socio-technical Systems

In his text book on software engineering Sommerville explains that software is embedded into a broader type of system which he calls „socio-technical" and defines as follows [Sommerville (2004), p. 21]:

> *„Socio-technical systems include one or more technical systems but, crucially, also include knowledge of how the system should be used to achieve some broader objective. This means that these systems have defined operational processes, include people (the operators) as inherent parts of the system, are governed by organizational policies and rules and may be affected by external constraints such as national laws and regulatory policies."*

This definition is more detailed than the previous ones and mentions aspects such as the usage of the system, operational processes, organizational policies and rules, which are important for CSCW-projects; they will be included my conceptualizations.

What is problematic about Sommerville's conceptualization is the methodological support. He takes a very technical view on socio-technical systems by defining a discipline that is responsible for socio-technical systems [ibid, p. 25]:

> *„Systems engineering is the activity of specifying, designing, implementing, validating, deploying and maintaining socio-technical systems."*

As in the definitions discussed above, it is implied that the social as well as the technical part of a socio-technical system can be designed by engineering methods. Thus also Sommerville's conceptualization falls short in providing a theoretical background for describing phenomena occurring during the appropriation of software by organizations.

2.1.4 Other Conceptualizations of "Socio-technical Systems"

There are two other conceptualizations of "socio-technical systems" that need to be discussed. Both are founded on system theory but they differ from each other in fundamental concepts.

The first author to mention is Ropohl who elaborated a "general technology" including a conceptualization of "socio-technical systems" [Ropohl (1999), pp. 135]. He derives his concept of "socio-technical systems" from the concept of division of labor, first between humans and then between humans and machines. For Ropohl "socio-technical systems" are created when there is an activity system serving a defined goal and in which some activities are carried out by humans and others are carried out by machines [ibid. p. 141]. The problem with his conceptualization is that he assumes an equivalency between processes carried out by machines and processes carried out by humans. By failing to describe the differences between processes inscribed into technical systems and processes carried out by human actors, Ropohl's theory falls short for the purposes of this thesis in two respects: First, it does not provide the basis for deriving differentiated methodological support for organizational change on the one side and software engineering on the other. Second, it does not provide any concepts for describing the special relationship between social and technical systems. One aspect of this special relationship has been described as appropriation or evolving use in section 1.1.2. The empirical evidence about the complex processes of appropriation contradicts Ropohl's assumption that within a socio-technical system, social processes and processes inscribed into technical systems can be regarded as equivalent.

Another conceptualization was elaborated in [Loser (2005)][5]. Loser's goal is to describe socio-technical systems as autopoietic systems consisting of a network of homogeneous elements. Based on Luhmann's system theory, Loser defines "communications and controlling actions" as the basic element for socio-technical systems [ibid. p. 70]. The advantage of this form of conceptualization is that the new, emergent unit that is formed by a social and a technical system is explicated.

2.1.5 Motivation for a Reconceptualization

Why is a new Concept for "Socio-technical System" needed?

Why can a methodological support for CSCW-systems to integrate software engineering and organizational change not be based on the conceptualizations of "socio-technical system" summarized above?

There is not a single answer covering all conceptualizations introduced above. Each of them has their strengths and weaknesses. In general one can say that the early conceptualizations that resulted from research at the Tavistock Institute are strong in their theoretical foundation, but they are outdated in two respects: first, they refer to industrial production and machines used in this context some 50 years ago. Their methods do not refer to the special properties IT-systems in contrast to these older machines. Second, they do not include insights from newer system theory that have evolved in the 1980's. Ropohl does include IT-systems into his argumentation, but he refers to older systems and does not relate to the special social properties of CSCW-systems. Furthermore, he explicitly excludes concepts from newer system theory from his reasoning, and therefore misses the chance of describing social phenomena in theoretical terms.

The conceptualizations in the context of IT-systems and software on the other side do not revert to any system theoretic background; thus they are weak in their theoretical foundation. Methodologically they lack a clear interface to social theories and methods; they rather attempt transferring engineering methods to organizational change. Their strength however is that they document the interdependency of technical and organizational aspects in IT-design.

Loser's approach is closest to my reconceptualization. He described how technical and social systems can relate to each other such that a new emergent socio-technical system evolves. He concentrated on explaining the structure of the socio-technical system using the concepts of penetration and interpenetration [ibid, p. 71]. I will take a closer look into the dynamics of the special relationship between organizations as social systems and CSCW-systems as technical systems for deriving methodological support for CSCW-projects. My conceptualization is based on the concept of structural coupling, because it allows using proven methods for organizational change as well as for software engineering and combining both within one CSCW-project.

[5] Kai-Uwe Loser and I are members of the same research group and have worked together in several projects which provide the background to this thesis.

In the following the two main motivations for a reconceptualization are discussed in more detail.

"Open" vs. "closed" Systems

Because systems theory in the 1950s and 1960s was dominated by *"the notion of open systems that interact with their environment"* [Mumford (1996), p. 68], the concept of socio-technical system stemming from the original Tavistock research also includes this notion ([Emery, Trist (1960), p. 94][6], [Cummings, Srivastva (1977), p. 84]). Describing systems as "open" allows explaining their interrelation with their environment, especially the aspect that systems depend on their environment in many aspects of their existence, and that systems consequently react to changes in their environment. A biological system (e.g. an animal or a human being) can only exist if its environment meets demands concerning temperature or availability of food. If these conditions are not given anymore, the system must change (adapt) or will die. The same applies to social systems such as companies. If the market for their product is not available anymore, the company has to change by inventing new products or it will vanish.

While the notion of "open" systems can explain the interaction between a system and its environment ([Emery, Trist (1960)][6], [Cummings, Srivastva (1977), pp. 49]), it reaches its limits when it has to explain why social systems react very differently to identical changes in their environment. It is not possible to derive a function relating changes in the environment to changes in the social system. From this perspective, the social system is "closed". Yes, it reacts to changes in its environment, but it does so according to its own rules which cannot be determined from the outside of the system. This is the basic idea of Luhmann's system theory in which social systems are described as operationally closed systems. My reconceptualization of "socio-technical systems" is based on Luhmann's system theory and makes use of his concept of operationally closed systems for explaining phenomena occurring in the context of CSCW-projects and for preparing the design of methodological support.

Problems of Approaching Social Systems with Engineering Methods

The conceptualizations of „socio-technical systems" in the context of IS development seem to view the „social side" as some kind of add-on to the technical side. With regard to methodology, there seems to be a belief that technical methods can be extended to reach into the social sphere. That this is not as easy as it may seem has been recognized early in the Tavistock's research. The specific challenge of organizing „socio-technical system" was seen in their heterogeneity (Vaill cited in [Cummings, Srivastva (1977), p. 55]):

> *„To make the system work, the problem is to get the things [the technology] whose behavior is governed by one set of laws, to interface effectively with the men [the social system], whose behavior is governed by another set of laws."*

The effects that have been described as metamorphosis or evolving use (cf. section 1.1) are indicators that engineering methods that work well for the technical part of a „socio-technical system" are not appropriate for addressing the social part.

[6] The referenced article is identical with [Emery, Trist (1969)] which might be easier to obtain.

In the course of CSCW-projects, often the need arises to decide whether a certain aspect of the work procedures should be ensured by technical or organizational means. Consider the following example taken from the second case study which is described in section 6.3. A team of university researchers designs a CSCW-system named ELISE for supporting their reading of scientific periodicals. One purpose connected to the system is that the researchers should scan the table of contents of new journals on a regular schedule such that they are aware of new articles and that the student worker responsible for ordering articles does not have to go for the same journal more than once. Technical options for ensuring a regular schedule are reminders e.g. by automated e-mails or entries in the group's calendar. Other technical options would be to remove journals from the list of available journals after some time such that no more orders can be placed and thus forcing the researcher to look at them before they are deleted. Organizational options include commitments by the team members, dedicated time for joint sessions in the system, and personal reminders by the library staff. In any case, the researchers have to adjust their weekly schedule to include regular reading of journals by using the ELISE.

All options are feasible and can be combined but technical and organizational options require quite different means for their realization. The technical options are transformed from requirements to specifications and implemented into the CSCW-system. The organizational options need to be transformed into conventions or rules within the organization which then have to be followed by the organization's members. The following aphorism (Lorenz' chain), which is attributed to the behavioral biologist Konrad Lorenz and often used to illustrate the problems inherent to organizational change, summarizes the uncertainties in implementing organizational rules:

> *„Said is not listened, listened is not understood, understood is not agreed, agreed is not kept in mind, kept in mind is not applied, applied is not maintained!"*

Transferred to CSCW-projects, this implies that the establishment of work-procedures needed to complement a CSCW-system cannot be engineered the way the technical system can. My reconceptualization of "socio-technical systems" prepares the ground for relating proven methods of software engineering to proven methods for organizational change rather than extending engineering methods into the field of organizational change.

The following sections discuss the two "*independent but correlative*" [Cummings, Srivastva (1977), p. 60] parts of a socio-technical system in the light of newer system theory: the technical system (cf. section 2.2) and the social system (cf. section 2.3). Section 2.4 introduces the original concept of self-descriptions and their role in processes of organizational change.

2.2 The Technical Side of „Socio-technical Systems"

2.2.1 Typology of Systems according to Luhmann

Figure 2 illustrates the hierarchy of different types of systems according to Luhmann [(1995), p. 2]. Machines are technical systems; human beings consist of several systems including a biological and a psychic system; social systems emerge from the social behavior of psychic systems and organizations are one form of social systems. For this thesis, the types “machines” and “organizations” are the most important ones. While this section regards software systems as a special form of technical systems, section 2.3 introduces Luhmann’s theory of social systems in general before focusing on organizations.

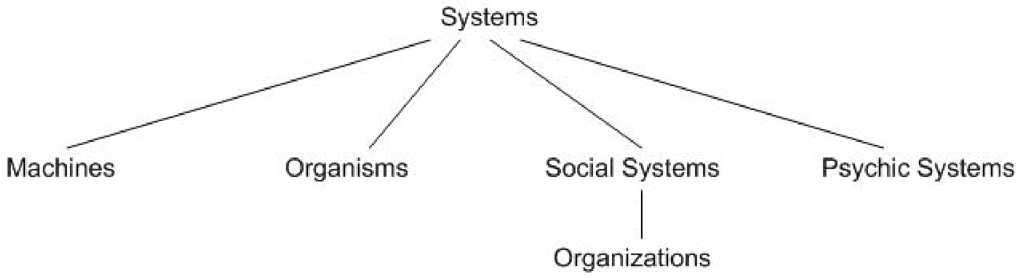


Figure 2: Typology of Systems

(Adapted from [Luhmann (1995), p. 2])

2.2.2 Technical Systems in General

The property of technical systems that is relevant for this chapter is that they are man-made and that they serve a specific purpose. This is the very basis of any engineering method – including software engineering. To cite Sommerville again [Sommerville (2004), p. 7]: *„engineers make things work”*. Engineering methods in general contain instructions for the planful design and implementation of technical devices.

Man designs and creates machines (or software) for a certain purpose and as long as the engineering is done correctly, a machine will do what it has been designed for. Any erroneous behavior of the machine can be tracked down to a mistake either in design or in implementation. Within systems theory systems of this type are referred to as allopoietic [Kneer Nassehi (2000), p. 49]. „Allopoietic” comprises the Greek words „allo” and „poiein”. „Allos” means „other”[7]; „poiein” means „create”. Thus, technical systems are characterized as „being made by others”. Social systems on the other hand will be described as „autopoietic”, meaning „self-creating” (cf. section 2.3.2); and this distinction between allo- and autopoietic is one strong argument for differentiating between technical and social systems when elaborating methods for CSCW-projects.

Another characterization of technical systems will be relevant for the further discussion: technical devices reduce the complexity and contingency of human action. A workflow management system, for example, foresees certain steps to be taken in a certain order. Thus it embodies rules in an inflexible way; in his actor-network-theory Latour uses the

[7] "all-." Webster's Third New International Dictionary, Unabridged. Merriam-Webster, 2002. http://unabridged.merriam-webster.com (25 Aug. 2004).

term „inscription" for social rules that are fixed in technical devices. This aspect will be taken up again in section 3.1.3 (p. 73).

Luhmann defines technics as the *„direct coupling of causal elements"* [Luhmann (2000), p. 364]. This definition includes the aspect of limited contingency which Robinson refers to as "operational predictability" for his concept of "common artifacts" ([Robinson (1993b), p. 158]; cf. section 3.1.3, p.71). Luhmann further differentiates on the basis of the types of operations that are coupled. In the case of physical or chemical operations the resulting systems are technical machines.

Different types of technical devices (e.g. hard- and software) are combined for computer supported cooperative work. The focus of this thesis is on the design of interactive software supporting cooperative work and the following sections provide some basic definitions needed for the further discussion.

2.2.3 Software Systems and Software Engineering

For the further discussion, a definition of software and software engineering is needed. Definitions that capture widely accepted interpretations of the terms can be found in Sommerville's textbook on software engineering; I will use them within this thesis.

> *"A **software system** usually consists of a number of separate programs, configuration files which are used to set up these programs, system documentation which describes the structure of the system and user documentation which explains how to use the system and, for software products, web sites for users to download recent product information."*
>
> [Sommerville (2004), p. 5]

Term 5: Software-System

The matching definition for software engineering that I adopt for this thesis is also taken from Sommerville

> *"**Software engineering** is an engineering discipline which is concerned with all aspects of software production from the early stages of system specification through to maintaining the system after it has gone into use."*
>
> [Sommerville (2004), p. 7]

Term 6: Software Engineering

There are valid arguments that software is more than a technical system and that software engineering is not merely a technical discipline [Floyd, Züllighoven (2002), p. 765]. The topic of this thesis itself is motivated by the close relationship between software engineering and organizational processes. However, in order to describe possible integrations of software engineering and organizational change, I start with widely accepted definitions of software engineering and software before identifying and elaborating the options for integration with organizational change. An indication that the definitions given in Term 5 and Term 6 are widely accepted is that both German (Gesellschaft für Informatik) and international (ACM, SigSoft) associations of computer science provide similar interpretations ([Webpage GI] [Webpage ACM SigSoft]).

2.2.4 CSCW-Systems

Since this thesis is not concerned with software in general but with the specific kind of software-systems that are designed and employed for supporting cooperative work, two definitions that are in line with the hierarchy provided by [Prinz (1993), p. 151] shall be added for substantiating the CSCW-related discussion.

> The term **CSCW-System** shall be used in this thesis to denote a technical system including hard- and software that provides special features for the support of cooperative work.

Term 7: CSCW-System

> The term **CSCW-Application** shall be used in this thesis to denote a software system that provides special features for the support of cooperative work. A CSCW-application is part of a CSCW-system.

Term 8: CSCW-Application

Instead of CSCW-application, the term CSCW-software could be used, but CSCW-application has been used in the field of CSCW before [Prinz (1993)] and therefore has the advantage of being connectable to other discussions. Another alternative would be the term „groupware" which is often used as a synonym for CSCW-application. But, as Schmidt and Bannon point out, cooperative work is not limited to groups [Schmidt, Bannon (1992), p. 17]. The term group is used in social sciences and psychology for describing a number of people who

- are united by a sense of belonging together,
- share values and norms, differentiate roles,
- identify with a common goal or task,
- are set apart in time or space from their other environment,
- interact with each other

[Sader (2001), p. 39]. There are settings of cooperative work in which the participants do not form a group and these settings should not be ruled out from the scope of this thesis. Indeed, researchers often use the term groupware in a broader sense referring to

IT-systems that do not require that their users form a group; but in order to avoid associations which may be invoked by the term group, I will use the term CSCW-application instead.

These definitions of CSCW-system and CSCW-application are also corresponding to the idea that a software system is embedded into a larger technical system that includes hard- and software [Sommerville (2004), p. 25].

What are software systems that qualify as CSCW-applications in the sense of Term 8? According to the goal formulated for the field of CSCW in Term 1, CSCW related research addresses all those software systems that are used for and within cooperative work arrangements. It is not a specific technical characteristic that qualifies software as a CSCW-application, but rather its ability to support cooperative work (cf. [Teufel et al. (1995)], [Schwabe, Streitz, Unland (2001)], [Andriessen (2003), pp. 10] for overviews). Typical examples of CSCW-applications that have been subject to research include:

- Group calendars (e.g. [Grudin (1988)])
- Group editors (e.g. [Gutwin, Greenberg (2002)])
- Group decision support systems (e.g. [DeSanctis, Poole (1994)])
- Project management (e.g. [Grudin (1988)])
- Workflow management (e.g. [Herrmann, Hoffmann (in print)])
- Shared Information spaces and knowledge management systems (e.g. Orlikowski (1996)])

There are special technologies that have not (yet) been subject to CSCW related research but may be used for the implementation of CSCW-systems. Software systems employing methods of knowledge representation and reasoning, so-called knowledge based systems (KBS) [Brachman, Levesque (2004)] are one example. Knowledge representation and reasoning are known to be used for expert systems in a single-user environment (e.g. systems like the historical MYCIN). But KBS can also be used in cooperative work settings: case-based reasoning systems for example can support work in lawyers' offices and then they share many characteristics with knowledge management systems as they have been regarded in the field of CSCW [Beierle, Kern-Isberner (2000), pp. 152].

Now that the technical side of „socio-technical systems" has been substantiated with respect to CSCW-projects, Luhmann's newer systems theory shall be used to do the same for the social side.

2.3 The Social Side of „Socio-Technical Systems"

2.3.1 Sources for Luhmann's Theory

Referring to Luhmann‘s system theory in any scientific writing poses the problem to provide an adequate recapitulation of his complex oeuvre. Many texts that summarize and explain his theory are available and simply adding to these by providing yet another comprehensive summary cannot be the aim. However a basic understanding of his concept of social systems as autopoietic systems is necessary in order to understand the concepts I elaborate in this thesis. Although addressing an interdisciplinary audience, this is a thesis within the field of computer science and hence many readers may not be familiar with sociological system theory. Therefore the purpose of the following sections is twofold:

a) To provide enough information for a basic understanding of the concept of social systems as autopoietic systems.

b) To identify its potential for explaining and designing CSCW.

Luhmann‘s book „Social Systems" is the most important primary source [Luhmann (1995)][8] for this chapter. „Organisation und Entscheidung"[9] [Luhmann (2000)] is referenced as it explains general statements about social systems in the special context of organizations; „Einführung in die Systemtheorie" [Luhmann (2002)] is a transcript Dirk Baecker produced from a lecture Luhmann held at the University of Bielefeld in the winter of 1991 / 1992. Helpful summaries and additional explanations can be found in [Kneer, Nassehi (2000)] and [Baecker (2002)]. Rare primary sources in English include [Luhmann (1986)] and [Luhmann (1990)].

2.3.2 Social Systems as Autopoietic Systems

Social Systems Consists of Communications

This section starts with a definition of social systems. Within systems theory organizations are considered as one special kind of social systems; therefore everything that is true for social systems in general also applies to organizations. The characteristic features of organizations that distinguishes them from other social systems such as families or societies are discussed in section 2.3.6)

Luhmann‘s definition of a social system often gives rise to surprise or even rejection: social systems are not composed of persons but of communications. If one engages with this definition, two aspects will become apparent:

1. This definition allows a clear theoretical description of the characteristics in which social, psychic and technical systems differ from each other.
2. The first impression that individuals are irrelevant when it comes to explaining social phenomena is not accurate. As social systems can only exist in interaction with their environment, and as psychic systems are part of that environment, they do influence the creation and reproduction of social systems significantly.

A definition of social systems in the sense of Luhmann can be found in [Kneer, Nassehi (2000), p. 80]:

[8] It is a translation of [Luhmann (1984)]

[9] "Organization and Decision"

> *„Social Systems are autopoietic systems, which produce communications from communications in a recursively closed process."*

So communications are the basic elements of social systems. The definition also states that social systems reproduce themselves in a recursively closed process: Communications allow other communications to follow which in turn are the bases for further communications and so forth. Hence communications continuously connecting to other communications make up a social system.

Explanatory Potential of the Concept

This definition helps to explain how the life-span of social systems can exceed that of individuals by far. A company is perceived as the same company over years although it is likely that many employees have been exchanged in the meantime. Rather than the individuals, the communicative structures are what differentiate the company as a social system from its environment. Another illustrative example of a social system that has outlived many generations of men is the Catholic Church. This definition also helps to formulate a challenge with regard to CSCW-projects: the usage of the CSCW-system should be interwoven with the communications of the social system such that references to it are integrated in the process of self-making. If one succeeds in doing this, the effect that a CSCW-system is designed, implemented, rolled out but not embedded into corresponding work-procedures could be avoided.

Autopoiesis

The term „autopoietic" which is used in the definition was coined by the Chilean biologists Maturana and Varela[10]. It contains the Greek syllables „autos" meaning „self"[11] and „poiein" meaning „create". Within system theory those systems that (re-)produce their own components from their own components are called autopoietic.[12] In the case of social systems communications continuously produce new communications. Luhmann uses the term self-referential system more often but in the same sense as autopoietic system. [Luhmann (1995), pp. 32].

Another remarkable aspect of the definition is that social systems reproduce themselves using a „recursively closed process". Luhmann abandons the concept of open systems which was pursued in system theory for a many years and introduces the concept of self-referentially closed systems. Social systems are operatively closed in the sense that communications can only be brought forth from previous communications; in a

[10] cf. [Maturana, Varela (1987)] for a complete description of their concepts; or [Varela (1981)] for a condensed discussion on the issue

[11] aut-. Webster's Third New International Dictionary, Unabridged. Merriam-Webster, 2002. http://unabridged.merriam-webster.com (25 Aug. 2004)

[12] Maturana and Varela developed and used the concept of Autopoiesis for describing the distinct features of living systems; Luhmann then transferred it to social systems. From this history it is sometimes concluded that Luhmann defines social systems in analogy to biological systems. Luhmann himself explicitly negates this ([Luhmann (1995), p. 14] or [Luhmann (1984), p. 32]). His goal is to generalize Maturana and Varela's concepts in form of a general system theory and then to specialize them again relating to the properties of social systems.

continuous process of renewal communications refer to older communications and thereby enable future communications to take place. This „recursively closed process" is the process of Autopoiesis in which a social system creates itself by using its own components.

Autopoietic vs. Allopoietic - Explanatory Potential

The terms and concepts of system theory discussed so far can be used to describe differences between social and technical system. What has been vaguely referred to as „different set of laws" by Vaill (cf. section 2.1.5) can now be described more precisely using the terms of system theory: Technical systems do not reproduce themselves, instead they are made by someone or something other than themselves and called allopoietic (cf. section 2.2.2). Social systems can be described as autopoietic, meaning that they are not produced by anything external, but that they continuously reproduce themselves.

By describing social systems as autopoietic, phenomena such as the evolving use of groupware can be described and explained: while the CSCW-application as a technical system can be designed and implemented, the adoption of the technical system by the social system is an autopoietic process within the social system and cannot be engineered from the outside.

Perturbation

From all that has been said so far, one could conclude that a social system is not influenced by its environment at all. This is not the case. While the flow of communications within an organization cannot be determined from the outside, the environment can perturb a social system; i.e. it can give impulses. The reaction to this perturbation however is determined by the way communications connect to each other within the social system. Methods of systemic intervention (cf. section 4.2) try to perturb social systems in certain ways in order to invoke likely reactions. For CSCW-projects, the question is: What kind of perturbations can support an organization in the process of including a CSCW-system in its network of communications?

Since communication is such a central concept in the description of social systems and since many definitions exist for this concept, it is worthwhile to look at Luhmann‘s definition of communication.

2.3.3 Communication as a Three-Part Unity

Three Selections form one Act of Communication

According to Luhmann communication is the synthesis of three selections ([Luhmann (1984), p. 195] or [Luhmann (1995), p. 140):

1. Information: The utterer selects the information that he wants to convey. Luhmann calls this person („alter").

2. Behavior: In a second selection the utterer chooses a behavior that conveys the information.
3. Understanding: The addressee (Luhmann calls him „ego") performs the third selection by differentiating between the utterance and the information.

If and only if all three selections take place, it is that Luhmann talks of communication [Luhmann (1995), pp. 141-142]. Figure 3 illustrates how one act of *Communication* consists of three sub-activities.

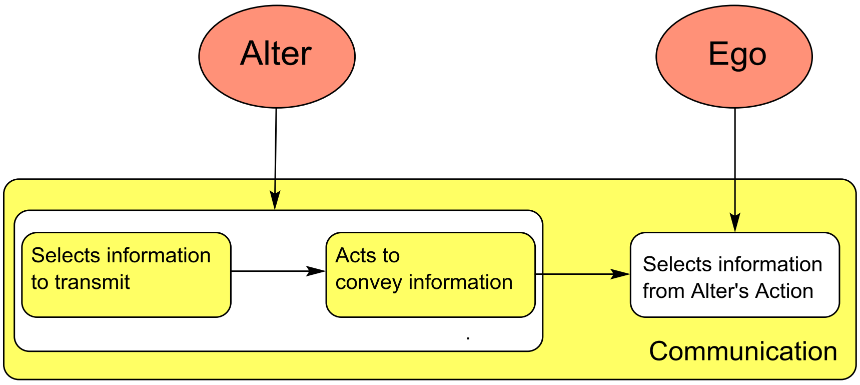

Alter
Ego
Selects information to transmit
Acts to convey information
Selects information from Alter's Action
Communication

Figure 3: Communication as a Three-Part Unity

The following example taken from the context of logistics should illustrate the definition: A truck driver wants to inform his dispatcher that he has completed unloading at a customer's site and that he now continues his tour (selection 1: information). The truck driver picks up his cell phone, calls the dispatcher and says: „Everything ok at Meyer's, I am on the road again." (selection 2: behavior). The dispatcher understood that the order „Meyer" has been completed and that the driver is now on his way to the next customer (selection 3: understanding).

Because communication is a central concept for this thesis and because there are many different understandings and definitions of this concept (cf. section 3.1.4, p. 78), it is worthwhile to include an explicit definition at this point.

Communication is defined in the sense of Luhmann ([Luhmann (1984), p. 195] or [Luhmann (1995), p. 140) as an interaction between two persons (alter and ego) comprising the three selections information, behaviour and understanding. Figure 3 illustrates the definition.

Term 9: Communication

What makes communication so difficult in everyday life is the phenomenon of contingency which is discussed in the next section.

Contingency and Double-Contingency

The term contingency denotes *„the condition that something may or may not occur"*[13]. Luhmann uses the term contingency in the sense of *„also being possible otherwise"* ([Luhmann (1995), p. 25], [Luhmann (1984), p. 47]) and argues that any act of selection implies contingency. Since communication consists of three selections, it is a contingent event: The driver in the example above could have selected different information to tell the dispatcher and the dispatcher could have selected a different understanding from the information he received. Luhmann uses the term double-contingency is to describe the situation of mutual contingency: alter cannot determine the selections ego makes and vice versa.

This is the emergent property that Luhmann attributes to communications: no single person can determine a flow of communication; any communicative act depends on selections made by at least two individuals and its outcome is therefore not predictable for the individuals.

This contingent characteristic of communication is the fundamental difference to technical systems the way Luhmann defines them (cf. section 2.2): technical systems consist of firmly coupled operations while social systems have to cope with the phenomenon of double contingency. Although techniques such as evolutionary algorithms can be regarded as contingent, CSCW-systems are not designed to be contingent in their behavior. On the contrary - the main purpose of introducing a workflow-system, for example, is to ensure the „direct coupling" of events by technical means. So within a CSCW-project two types of systems are relevant: social systems which are based on the contingent phenomenon of communication and technical systems that are designed to ensure a direct coupling of events.

The contingent nature of communication has another implication for CSCW-projects. Lorenz' chain was cited above (cf. p. 32) to illustrate the difficulties in supporting organizational change processes. The description of communication as a contingent phenomenon is a theoretical construct with the same implications: it is not sufficient to state that some work-procedure should be performed in a certain way. There is no guarantee that the other(s) will interpret the information in the way the speaker meant it; and even if they do so, there is no guarantee that they will realize it. These insights about social systems let the notion of „engineering" „socio-technical" systems (cf. section 2.1.3) appear questionable. The term „engineering“ bears the connotation that it is possible to define (engineer) some work-procedure once and then it will undoubtedly be „implemented" by human actors.

2.3.4 Communication and Action

Communication is often attributed to one person by identifying the three-part unity with the single act of uttering (selection 2: behavior). Actions such as the act of saying something can be observed and described.

[13] Webster's Third New International Dictionary, Unabridged. Merriam-Webster, 2002. http://unabridged.merriam-webster.com (29 Aug. 2004)

> *„Therefore it is not false, only one-sided, for a communication system to interpret itself as an action system. Only by action does communication become fixed at a point in time as a simple event."*

([Luhmann (1995), p. 165] or [Luhmann (1984), p. 227]). The relevance of this theoretical concept for CSCW-projects is twofold: First, it shows that viewing organizations as social systems consisting of communications is compatible with standard practices of describing organizations in the context of CSCW-projects. Second, a first reference is made to what will later become the concept of self-description. Luhmann says that social systems interpret themselves and that they use action as a basic element to do so. This will become important in the course of this thesis when the content and form of socio-technical self-descriptions are discussed.

But before turning to self-descriptions in section 2.4, some other concepts need to be introduced in order to unfold the explanatory potential that system theory provides for phenomena occurring in CSCW-projects.

2.3.5 Individuals, Social and Technical Systems

In this account of Luhmann's theory of social systems the individual person has not yet been put in place. For CSCW-projects this issue is relevant because both, the individual use of a CSCW-system as well as the communicative (i.e. social) use, will be discussed (cf. section 3.1.4).

If human beings are not the constituents of social systems, but if at the same time their influence on social processes is not denied – what exactly is their position within systems theory? According to Luhmann human beings are no autopoietic unity but comprise several different systems [Kneer, Nassehi (2000), p. 80]. The psychic system is one of these.

Psychic Systems

Psychic systems are autopoietic systems that are based on conscious states ([Luhmann(1995), p. 59] or [Luhmann (1984), p. 92]); they belong to the environment of social systems. The network of operations that make up psychic systems consists of thoughts; any thought in a human mind is the result of a previous thought and gives itself rise to new thoughts. Psychic systems cannot exist without social systems and vice versa; their relationship is so special that Luhmann uses the term interpenetration to describe it: The first selection of the three-part unity communication requires a psychic system to select the information to be conveyed. This selection is performed within the network of thoughts that constitute the psychic system; by adding the other two selections (behavior and understanding) communication emerges. Thus although the autopoietic networks of psychic and social systems are both closed in themselves, they are intertwined at the same time to enable communication[14].

[14] Psychic systems themselves rely on the perceptive abilities of biological systems

The Relation of IT-Systems, Psychic Systems and Social Systems

Concerning the design of IT-systems both the individual level of psychic systems and the emergent level of social systems are relevant. Methods developed in the field of HCI (human computer interaction) aim at the individual user. Ergonomics treat the question how to design an interactive system in order to support an individual best in task completion; concepts for qualification aim to enable an individual to make the best use of a software-system; and motivational issues are included to ensure that individuals are willing to engage with a new system. The field of CSCW (Computer Supported Cooperative Work) however focuses on the needs of groups[15] and how they coordinate themselves to accomplish a task. Here the level of the individual is not sufficient anymore to explain all phenomena.

If, for example, all members of a team know very well how to put information into a knowledge management system, it is not ensured that they create a knowledge base that is as helpful as everybody expected. The first step – the knowledge about how to use the system – takes place on the individual level; the second step however – the agreement on what information should be entered when – takes place on the social level. For the second step communications are needed among the members of the organization who will use the KMS. Each individual user has his or her own ideas about the usage of the KMS; it is necessary to share and discuss these ideas in order to come to a common idea on the social level.

Perception of Technical Systems by Social Systems

How can a social system relate to a technical system in its environment? Since communication is the sole operative base of a social system, it must be done using communicative acts.

> *„On the level of this self-referential organization, self-referential systems are closed systems, for they allow no other forms of processing in their self-determination."*

[Luhmann, (1995), p. 34]. This description of social systems as operatively closed systems implies that for a social system only those things exist that are expressed in form of communication. And any event in the environment needs to find its way into the network of communications in order to be of any relevance for the system's behavior. This also applies to CSCW-systems: as far as a social system - such as a group of people working together - is concerned, the technical system only exists if it is part of the communications. Even if all members of a group are perfectly qualified and willing to use a CSCW-system, it will not become part of their cooperative processes unless they communicate about it.

Conclusion

If a CSCW-project aims at embedding a CSCW-system into the work-procedures of an organization, then methods need to be applied that ensure that the CSCW-system crosses the border from individual use to the social level. The theoretical concepts

[15] 'group' in this context does not necessarily imply a group in a sociological sense; i.e. people sharing common norms and having a common identity; cf. [Schmidt, Bannon (1992)]

discussed so far give indications how this could happen: the organization needs to communicate about the CSCW-system. This is indeed the core of the concept and techniques I suggest: within a CSCW-project, the organization should be supported in communicating about the CSCW-system and its utilization.

2.3.6 Organizations as a Special Form of Social Systems

Organizations consists of „Decisions"

So far only social systems in general were regarded. But how does systems theory differentiate between different types of social systems? A family, a company, three friends watching a game of football on TV, ten-thousand fans in a stadium, and Germany as a society, are examples of very different social systems. Luhmann distinguishes between them by describing the specific kind of communication that forms the respective systems. The type of social systems that is relevant within this thesis is the organization, and the type of communication specific to organizations is „decision" ([Luhmann (2000), p. 63], [Willke (1994), pp. 150]).

Often organizations are characterized by their members. Not the individual persons but the roles that are necessary to maintain the organization. In a school for example there are teachers, students and parents. Roles are usually characterized as bundles of interrelated expectations or a set of rules that determine tasks to be performed be the role-taker. Luhmann does not take roles to be the building block of organizations but dissects them one step further. Expectations and rules are a combination of information and demand which are in turn aspects of communication. Expectations connected with a role in an organization can be so precisely phrased that they can be used to determine whether a certain communication belongs to the organization or not. A role-taker decides with each communication whether he adheres to the rules of the organization or not. In this context Luhmann regards communication as decision *„if and insofar as the slant of meaning an action has is in reaction to an expectation directed to that action."* ([Luhmann (1995), p. 294] or [Luhmann (1984), p. 400]).

At this point the special type of communication that makes up organizations is available: organizations consist of communicative acts that react to expectations. This type of communication is called decision within Luhmann's theory.

Consequences for CSCW-Projects

How do these considerations link to CSCW-systems? First it can be stated that they are compatible with concepts used in CSCW-systems. Workflow systems, for example, send out mails to inform users that it is time to perform a certain task; i.e. the expectations towards the user are transported by the workflow-system. Many CSCW-systems foresee different roles for different users; connected to theses roles are expectations the other users may have towards the role-taker.

Second, the definition of organizations as networks of communications that react to expectations, gives hints for the processes of organizational change within a CSCW-project. The theoretical conceptualization implies that in order to change an organization, the rules that are effective in the organization have to be addressed rather than the individuals working in the organization [Willke (1994), p. 154]. If one takes this thought one step further to the deployment of a CSCW-system in an organization, it follows that it is not sufficient to qualify individuals in using the software. Rather, the rules that determine the flow of communication within the organization need to be changed such that they include the utilization of the CSCW-system. But how can an organization change its rules? Within the conceptualizations of system theory, the only way to do this is to communicate. So the conclusion of the previous section can be stated more precisely: the organization needs to communicate about its own rules concerning the CSCW-system.

Coming back to the simple example of a knowledge management system above, the expectations that the organization's members have towards each other concerning the usage of the KMS must be explicated and negotiated. Only that subset of expectations that is included in the flow of communications within the organization has a chance of being fulfilled.

For the clarification of the relation between technical and social systems and an alternative conceptualization of „socio-technical" systems, some more basic terms are needed. But first, a typology of systems is included to summarize the concepts discussed so far.

Focus of this Thesis: Organizations

Organizations are the type of social systems that are of interest within this thesis; therefore the following sections and chapters will talk about organizations rather than social systems in general. This does imply that not all conclusions, methods and techniques I elaborate can be transferred to all scenarios of cooperative use of computer systems. Since cooperative work usually takes place within organizations, this seems to be an acceptable loss of generality.

It may not go unmentioned that this focus is not shared by all CSCW-researchers. There are studies e.g. of families coordinating themselves with the help of a shared calendar that aim at drawing conclusions for the design of groupware [Crabtree et al (2003)]. However, the transfer of insights from a family setting to an organizational setting is questionable, because the usage of shared artifacts within a family might be determined by variables that do not apply to organizations.

2.3.7 Structural Coupling

Systems consisting of Homogeneous Elements

With regard to existing definitions of „socio-technical" systems (cf. section 2.1.3) it strikes that Luhmann's typology does not include any type of system consisting of different types of subsystems. This is no coincidence, but part of his conceptualization of systems. Luhmann points out, that self-referential systems cannot comprise heterogeneous elements ([Luhmann (1995), pp. 39] or [Luhmann (1984), p. 67]):

> *„On the level of elements, self-reference means that these connect up by referring back to one another and that interconnections of processes thereby become possible. But this can*

> *occur only if the types of element are sufficiently similar. Therefore, to cite an extreme case, no system unity can exist between mechanical and conscious operations, between chemical operations and those that communicate meaning. There are machines, chemical systems, living systems, conscious systems, and (social) systems that communicate via meaning; but no system unities encompass all these at once."*

If concepts of newer systems theory are used as a theoretical foundation for supporting organizational change in CSCW-projects, the integration of technical and social systems into a „socio-technical" system is not compatible. So the motivation for a reconceptualization provided in section 2.1.5 is supplemented by yet another, a formal, theoretical argument.

Relating to other Systems by Structural Coupling

It is common sense that social systems such as companies do react to changes in their environment. But how does that fit together with the description of social systems as operationally closed systems? „Structural coupling" is the answer to this seeming contradiction. „Structural coupling" denotes a special relationship between two systems: Structurally coupled systems depend on each other but remain in each others environment at the same time ([Kneer, Nassehi (2000), pp. 62][16], [Luhmann (2000), pp. 397]). The concept of structural coupling succeeds the concept of „adaptation" in the theory of open systems. While adaptation implies a direct reaction that is caused by the environment, the concept of structural coupling emphasizes the autonomy of the social system. Within its own autopoietic organization the social system can modify its structures in order to relate to its environment. There are two extremes that limit the process of structural coupling: on the one side the structural coupling must not dissolve the autopoietic organization of the system. Any change performed must preserve the autopoietic processes or else the autopoiesis will cease and the system will dissolve. On the other hand no autopoietic system is independent from its environment. The autopoietic processes rely on exchange with their environment and therefore the system needs to be flexible in its structural coupling in order to survive in changing environments.

Since the concept of structural coupling is of central importance for this thesis, it is captured as a term below:

The term **structural coupling** denotes a special relationship between an autopoietic system (a) and another system (b) in its environment. System (a) adapts its inner structures such that it can relate to system (b). Thus, the existence and the characteristics of system (b) influence system (a) but they do not determine the way it adapts itself.

Term 10: Structural Coupling

[16] Cf. [Maturana, Varela (1987), pp. 75]

The concept of structural coupling is closely related to the concepts of adoption and appropriation known in the field of CSCW. A discussion of the three concepts is included in section 3.2.1 (p. 83).

Explanatory Potential of the Concept of Closed Systems

One might ask why the concept of open systems is abandoned in favor of the more complicated concept of closed systems, when even more concepts – such as structural coupling – are needed to describe the relation between systems. Willke answers this question by explaining that self-referential closure is the one mechanism that enables a complex system to maintain its inner organization in a turbulent environment [Willke (1994)]. So, from a sociological point of view the concept of self-referential closure is able to explain certain social phenomena better than other concepts can.

The phenomenon of evolving use introduced in section 1.1.2 can also be explained using these concepts. It was observed in [Andriessen, Hettinga, Wulf (2003), p. 367] that *"More often than not, users appear to use groupware in another way than the groupware designers intended or IT-departments expected."*. The concept of self-referential closure explains why the designers' intentions do not necessarily lead to the intended usage: When an organization deals with a new CSCW-system, the integration of the CSCW-system into the organization's procedures is determined within the organization itself by means of its autopoietic processes. If the software-system is considered important or helpful, a process of structural coupling will take place and the organization changes by adopting the CSCW-system. But this adoption does not necessarily coincide with the designers' intentions.

Now everything is available for discussing the concept of self-description which is of central importance for my socio-technical concepts presented in chapter 3.

2.4 Self-Descriptions in Organizations

2.4.1 General Characterization of Self-Descriptions

In contrast to other types of systems such as e.g. biological systems, organizations do not possess anything physical – like a membrane – that constitutes their boundary to the environment. Social systems must maintain their boundaries in a continuous process of negotiation deciding which communicative acts belong to the system and which do not ([Luhmann (1984), p. 268] or [Luhmann (1995), p. 196]). For this they need guidance and Luhmann introduces the concept of self-description as a means to serve this purpose. In a very intuitive sense, self-descriptions describe the main characteristics of the social system and thereby document the expectations that are directed toward its members.

Typical examples of organizational self-descriptions include

- Structural charts showing the affiliation of members to sections and their hierarchical order
- Business process descriptions showing how standard business cases are handled
- A mission statement describing the organization's purpose and goal
- Values and norms that are known and valid within an organization

The elaboration of the concept self-description is spread across the book „Social Systems"; additional explanations can be found in chapter 14 of „Organisation und Entscheidung" [Luhmann, (2000)]. Table 1 provides a selection of quotations taken from these sources that characterize self-descriptions from Luhmann's point of view. This table might serve as a reference for the reader interested in original quotes. For the purposes of this thesis, I derive four characteristics of self-descriptions that are discussed in the following sections. The right-most column indicates which quote contributed to which of the four characteristics.

	Quote	**Sd**
1	*„To make this possible, systems must create and employ a description of themselves; they must at least be able to use the difference between system and environment within themselves for orientation and as a principle for creating information."* ([Luhmann (1984), p. 25] or [Luhmann (1995), p. 9])	Sd3, Sd4
2	*„… self-observation must be carefully distinguished from the unity of the reproduction of the system's units (autopoiesis)."* ([Luhmann (1984), p. 61] or [Luhmann (1995), p. 35])	Sd2
3	*„Accordingly, self-observation is the introduction of the system / environment distinction within the system, which constitutes itself with the help of that distinction; self-observation is thus the operative factor in autopoiesis, because for elements to be reproduced, it must be guaranteed that they are reproduced as elements of the system and not as anything else."* ([Luhmann (1984), p. 63] or [Luhmann (1995), pp. 36])	Sd3, Sd4
4	*„But unlike nervous systems, structures and processes that employ meaning can include system boundaries and environments, which take on meaning within the processes of a self-referential system (not in themselves!), so that such systems can operate internally with the difference between system and environment."* ([Luhmann (1984), p. 64] or [Luhmann (1995), p. 37)	Sd4
5	*„All self-observation and self-description is ultimately a distinction, an operation of distinguishing."* ([Luhmann (1984), p. 105] or [Luhmann (1995), p. 69])	Sd4
6	*„Thus, a social system is constituted as an action system on the basis of communicative happenings, and using their operative means. The system generates a description of itself in itself to steer the continuation of the process, the reproduction of the system."* ([Luhmann (1984), p. 227] or [Luhmann (1995), p. 165])	Sd3
7	*„The continual production of individual actions within social systems can best be conceptualized as the performance of a concurrent self-observation by which elemental units are marked in a way that produces support points for further connective actions."* ([Luhmann (1984), pp. 229] or [Luhmann (1995), pp. 166])	Sd3
8	*„The consequence, at least for social systems, is that autopoietic reproduction and the operations of self-description and self-observation that use the system / environment difference within the system cannot be separated. The distinction retains its analytical value – but only to enable the hypothesis that social systems can carry out their self-reproduction only with the help of self-observations and self-descriptions."* ([Luhmann (1984), p. 230] or [Luhmann (1995), p. 167])	Sd2 Sd3 Sd4
9	*„Every self-description or self-observation by a social system is further communication and only possible as such (otherwise it would be a description or observation from outside, e.g. by an individual)."* ([Luhmann (1984), p. 232] or [Luhmann (1995), p. 168)	Sd1
10	*„Drawing on concepts from the theory of self-referential systems – namely, the idea that systems, by their own operations, can devise a description of themselves and then observe themselves – one can detach the connection among communication, action, and reflection from a theory of the subjectness of consciousness (the theory that consciousness must pertain to a subject)."*	Sd1

	Quote	Sd
	([Luhmann (1984), p. 234] or [Luhmann (1995), p. 170])	
11	*„Self-description is not only some kind of depiction leaving out the details, not only the outline of a model or a map of the self; to prove its worth, it must also increase the complexity it can experience by presenting the system as difference from the environment and acquiring information and guidance for connective behavior by means of this difference."* ([Luhmann (1984), p. 235] or [Luhmann (1995), p. 170])	Sd3 Sd4
12	*„Creating a description that reduces a social system to a connection between actions is a precondition of every observation that puts into play the difference between system and environment, that, for example, ascribes to the system characteristics that distinguish it from the environment."* ([Luhmann (1984), p. 247] or [Luhmann (1995), p. 180])	Sd4
13	*„Only what is made a theme in the system's communicative processes is valid as an internal observation (self-observation), because the system is accessible to itself only through communication. Observation by participating psychic systems that cooperate in the communication and contribute actions is already observation from without."* ([Luhmann (1984), pp. 247] or [Luhmann (1995), p. 180])	Sd1
14	*„As the communicable difference between action and observation, self-observation is the operation that underlies the formation of structures in the social system that produces them."* ([Luhmann (1984), p. 408] or [Luhmann (1995), p. 301])	Sd2 Sd4
15	*„… the hypothesis is that with a stronger differentiation between action and observation under the condition of the ongoing communication of self-observation, it becomes more probable that relatively improbable (more demanding, e.g., more specialized) functional orientations will take place and select corresponding structures."* ([Luhmann (1984), p. 408] or [Luhmann (1995), p. 301])	Sd2
16	*„A stronger differentiation between action and observation can be attained in at least two ways. … The first, more obvious possibility consists in differentiating roles for observers. … The other way of differentiating action and observation separates observation technically rather than by roles. It results from the technical expansion of possibilities of communication by writing and later by mechanical duplication (the printing press). Whether it legitimizes roles or not, written or printed communication forces a separation of action and observation because while reading one can hardly act or participate in the actions of others. Instead, one is set free to evaluate the communication being read and, in doing so, to observe. At first, recording what is read merely forms a content of consciousness. Any communication that follows from reading, however, is very likely to be different from what would have been offered by participants interacting in a situation, especially if readers can assume that their communication partners also read and have an understanding of the reality content of what they have merely read."* ([Luhmann (1984), p. 409] or [Luhmann (1995), pp. 301])	Sd2 Sd3
17	*„Writing therefore, initiates a structural development because it strengthens the basis for such development, the difference between action and observation. Not only is „more knowledge" at one's disposal, but structurally different arrangements and semantics for processing knowledge are formed, and in consequence themes for self-observation are opened up. Society, and in it many social systems, are given a much greater capacity for communicating self-observation without having to restrict or attenuate their capacity for action."* ([Luhmann (1984), p. 410] or [Luhmann (1995), p. 302])	Sd1 Sd2
18	*„If in the long run self-observation on the basis of a difference between action and observation crystallizes references to function and bases structural development on them, then this is an evolutionary process of „blind" variation and selection. Self-observation on the level of elementary communicational processes, namely, the infiltration of action observations back into communication, is not a process that an existing system comes to know better and better. For this reason it is a creative, morphogenetic mechanism, which scans events for their function and occasionally fixes the result in successful structural achievements. Their operation does not depend on anticipating its result. It does not guarantee that the formation of structures realizes the best ones possible or improves the lot of mankind."* ([Luhmann (1984), p. 411] or [Luhmann (1995), p. 303])	Sd1 Sd2
19	*„This rudimentary self-observation of the system on the level of its operations becomes self-description if it produces semantic artifacts to which further communication can refer and with which the system's unity is indicated. A clear differentiation of observation and description (and thus self-observation and self-description) comes about with the invention of writing. A description can be performed orally, but this presupposes a textual model developed on the basis of writing, in particular, long, disciplined texts whose understanding is largely independent of the situation. When in the context of such self-descriptions the participants speak of „we" or give their connection a name that can be spoken of in other contexts, this has entirely different consequences than when a self-observation is merely reproduced or an impression of presence is, so to speak, collectivized. Typically, self-descriptions create a meta-unification, an overestimation of coherence in observing*	Sd1 Sd2 Sd3

	Quote	Sd
	the system, and in this respect they can mislead external observers." ([Luhmann (1984), p. 618] or [Luhmann (1995), p. 456])	
20	*„Yet the self-observation of social systems can only be communicative occurrence; the psychico-concious observation of social systems by their participants is observation of others."* ([Luhmann (1984), p. 618, footnote 42] or [Luhmann (1995), p. 457, footnote 42])	Sd1
21	*„… for social systems (and perhaps for all systems that use events as elements) the question can only be in what direction a simplifying self-description or reflection guides reproduction. Deviant self-reproduction is unavoidable – such is life. But self-descriptions are selectively simplified and thus fix themselves contingently within a certain range of other possibilities, and this fixing may influence the system's development."* ([Luhmann (1984), p. 620] or [Luhmann (1995), p. 457])	Sd3
22	*„Under 'self-description' we shall understand the production of a text or a functional equivalent of a text (e.g. indexical expressions such as 'we' or 'here' or a proper name), with which and by which the organization identifies itself. The text does not need to be available as a biblical or canonical document; however whatever fulfils the function of self-description must fulfil certain requirements regarding the self-reference on system-level. It must span and integrate completely different situations, events, circumstances by identity of reference; it must unvarying denote 'the same' but be flexible at the same time regarding meaning."* ([Luhmann (2000), p. 417] translated from German)	Sd1
23	*„Obviously, a self-description can only be produced within the system itself. A description of the system by systems in its environment will always be an extrinsic description – e.g. when scientists develop theories about organizations or describe certain organizations as part of case studies. One must speak of „extrinsic descriptions" in these cases. There is one kind of activity (and of organizations) that cannot be categorized easily by this clear cut distinction: the process of organizational consultancy by external experts."* ([Luhmann (2000), p. 433] translated from German)	Sd1
24	*„They [self-descriptive texts] create the difference between compliance and deviation, and thereby allow the system to be provoked into deviation from its own text, if circumstances provide sufficient support."* ([Luhmann (2000), p. 419] translated from German)	Sd4

Table 1: Collection of Quotations pertaining to the construct Self-Description

These quotes alone are not concrete enough to support the elaboration of a method, therefore I derive four characteristics of self-descriptions (Sd1 – Sd4) which lead to the illustration in Figure 4 (Figure 4 uses the modeling notation SeeMe which is also used for the case studies; an overview of the notational elements is included in Appendix 11.2).

2.4.2 Form of Self-descriptions

Sd1: Self-descriptions take the form of both ephemeral communications and durable artifacts

Following the conceptualization of system theory, that social systems consist of communications (cf. section 2.3.2, p.37), self-descriptions must also be communications. This is an important insight because it raises the phenomenon of self-description from an individual to a social level; the ideas individuals may have about the organization are relevant for the organization's self-description if and only if they are made part of the communicative discourse within the organization (cp. Quote 9 or 13 in Table 1).

Consider some examples:

a. A scientific team that sits together and reviews how the internal cooperation concerning the selection and archiving of relevant articles from scientific journals is organized, is engaged in self-description.

b. The head of the same team who sits in his office and draws a process diagram representing the processes concerning the handling of scientific journals, creates a description that is external with respect to the team as a social system (cf. Quote 20 in Table 1).

c. If the head of the team presents his diagram to the group and a discussion about the diagram follows, then this discussion is a self-description again.

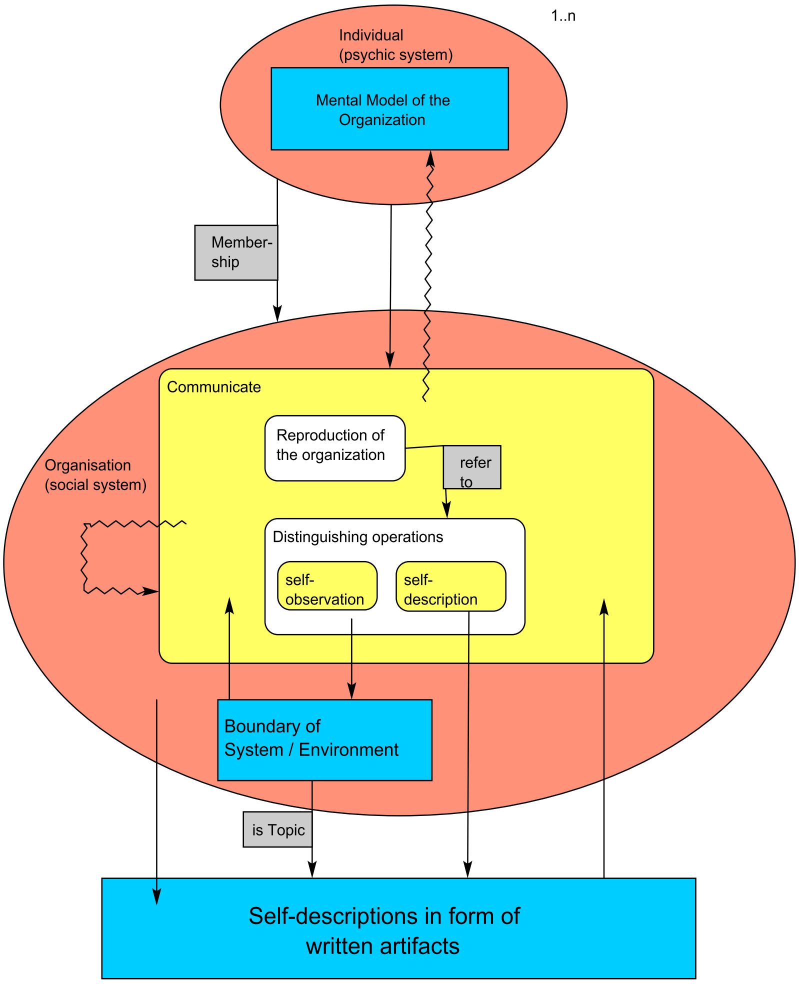

1..n
Individual
(psychic system)
Mental Model of the Organization
Member-ship
Communicate
Reproduction of the organization
refer to
Organisation
(social system)
Distinguishing operations
self-observation
self-description
Boundary of System / Environment
is Topic
Self-descriptions in form of written artifacts

Figure 4: Self-Descriptions in Organizations

In Figure 4 the organization is represented by a role that consists of a process *communicate*; the individuals that are members of the organization are located in its environment and connected via a relation labeled *membership*. These individuals carry out the process *communicate* and thereby create the complex relationship of interpenetration (cp. Section 2.3.5, p. 42).

Each individual creates his or her own mental model of the organization as a social system and these models are represented by the entity *Mental Model* within the role *Individual*. The fact that the *Mental Model* is embedded into the role *Individual* illustrates two aspects: first, that this *Mental Model* is an integral part of the *Individual* and thus influences the way the *Individual* communicates; and second, that this *Mental Model* is not directly reachable from the outside. However the individual *Mental Model* is influenced by the flow of communications within the organization and thus there is a feedback loop between an *Individual* participating in communicative processes and his / her *Mental Model* of the organization.

The communications that are self-descriptions of the social system are included within the activity *Communicate* as a subactivity *self-description*. Thus the aspect that self-descriptions are communicative acts is illustrated in Figure 4.

However, there is a second form in which self-descriptions can occur: written artifacts. In fact Luhmann points out, that self-descriptions can only serve their purpose if they are made durable in form of written artifacts (cf. quotes 17, 19, 21 in Table 1). Figure 4 includes the entity *Written Artifacts* that is a result of the activity *self-description* to illustrate this second form of self-description. Typical examples of written self-descriptions in organizations are organizational handbooks that include process descriptions for important business processes; or mission statements that include the organization's primary goal along with values that are held as important; or an organizational chart that shows the hierarchical structure and the association of members to specialized departments within the organization.

One last topic needs to be addressed in this context: the roles that participate in the process of an organization's self-description. At first sight, it seems to be quite clear: if a team talks about its way of working, it performs self-description; if the members create artifacts such as written text they make this self-description endurable. If an individual thinks about an organization, it is no self-description; even if the individual writes his thoughts down. But then, one can create examples that make the distinction less obvious: What, if only a few members of the team meet to talk about their way of working? Is that still self-description for the whole team? What, if the team engages a facilitator to support them in a structured discussion about their future work? Is that still self-description, although an external person was present and influenced the discussion? Similar questions may be asked with respect to written artifacts. Is an artifact that is produced by an external consultant (e.g. a process diagram) and that is then discussed and accepted by the organization a self-description?

Such questions become relevant in projects that aim at organizational change: organizations are often too large to allow each individual to equally participate in and contribute to processes of self-descriptions. Also, organizations often hire external consultants to support them. Luhmann recognized the problems connected to this setting (cf. quote 23 in Table 1) but did not give any practical advice to solve it; questions like the following remain open:

1. Who must be involved from within an organization if a communicative process shall be a self-description?

2. May external persons be involved in self-descriptive processes? If yes in which functions?
3. Who may be the author of written artifacts that shall serve the purpose of self-description?
4. How need such artifacts be re-included into communicative processes if they shall serve the purpose of self-description?

It will not be possible to give general answers to all of these questions but the methodological support for socio-technical self-descriptions will include guidance on these issues (cf. chapters 5 and 8).

2.4.3 Distinction of Self-descriptions

Sd2: Communications that serve the purpose of self-descriptions are distinct from other communications in the social system.

It has been said now, that self-descriptions are communications within a social system; another important statement is: not all communications within one social system are self-descriptions. The design of a method based on self-descriptions makes sense only if self-descriptions can be distinguished from other types of communication; if not, one should design a method for supporting communication in general.

Luhmann differentiates between communications by which the organization is reproduced and those communications that observe and describe the former (e.g. quote 2 in Table 1). Figure 4 includes all three types as subactivities of *communicate*: *Reproduction of the organization*, *self-observation*, and *self-description*. The latter are subsumed into one larger activity *distinguishing operations*. The zigzag arrow connecting *communicate* with itself illustrates how the *organization* autopoietically reproduces itself by building a network of communications.

The *distinguishing operations* are those communications that talk about the communications in the social system on a meta-level. They are called *distinguishing operations* because they distinguish between communications that are within the boundary of the social system and those that are not (cf. Sd4). The difference between *self-observation* and *self-description* is a gradual one (cp. quote 19). Both talk about other types of communications but *self-description* is the more endurable act that includes the creation of written artifacts. While *self-observation* is a necessary prerequisite for *self-description* it does not lie in a different category of communication. This thesis will concentrate on *self-description* as it is the stronger phenomenon. While it is important to be clear on the differentiation between *self-observation* and *self-description* on a theoretical level, it seems to be more important to elaborate the characteristics of *distinguishing operations* as a whole for a method supporting CSCW-projects. I use term *self-description* as a pars-pro-toto for two reasons: First, it seems to be the more intuitive term for practitioners. Second, the term has been established in this sense by methods of systemic intervention (cf. chapter 4.2).

So far it has been argued that it is possible to distinguish between *self-description* and other acts of communication. But Luhmann even reasons that the enforcement of such a distinction is beneficial for an organization. Quote 15 in Table 1 is of special interest in this context. It claims that those social systems that explicitly communicate self-descriptions are more likely to develop complex structures than those that do not. This is an important motivation for using self-descriptions in a method supporting CSCW-projects: if it were possible to support the appropriation of a CSCW-system in such a way that the organization gains abilities to develop complex structures in relation to the CSCW-system, much would have been gained.

While the distinction between self-description and „normal" communication is easy on an abstract level, difficulties occur if one tries to categorize concrete communications. Consider the following examples referring again to a scientific team organizing its procedures for selecting, ordering and archiving articles from journals:

a) A scientist sends a mail to a student worker in the library saying: I have read article xy, it is relevant for our project. Please get an electronic version of the article, store it in our database and give it the keyword xyz.

b) A scientist sends a mail to his colleagues saying: Because I was very busy last week and because we had the holiday, I am behind with my reading. Could we please postpone the deadline for ordering articles by two days?

c) The scientific staff sits together and discusses that the process of selecting and ordering articles is too slow. They decide that selected articles will be ordered each Thursday even if some members haven't had the chance to place orders.

a) is an example for a *reproduction of the organization*; i.e. a regular communication that keeps the organization alive. c) on the other hand is clearly an example for *self-description*; i.e. communication that talks about other communications on a meta-level. b) is something in-between: it is more than a regular communicative act within the boundaries of the organization, because it asks for an exception from rules. But it is less than self-description because it does not aim at permanently altering the boundaries of the social system.

For the design of a method supporting CSCW-projects it is important to find a closer definition of self-descriptions for two reasons: First, it needs to be clear which kind of communication should be fostered by a method enabling self-descriptions. Second, it should be possible for a given organization to determine the amount of self-description that takes place. While the first argument aims at designing methods for intervention, the second aims at measuring their success. The current chapter starts with the task of defining self-descriptions by deriving the four practical characterizations from Luhmann's theoretical work; chapter 3 continues by transferring the ideas to socio-technical systems and thereby bringing them closer to the subject of CSCW-projects. The case studies (chapter 6 and 7) will provide practical experience to be added.

One last issue to be mentioned in this section is that Luhmann does not only describe the difference between self-description and other types of communication but that he also suggests ways to enforce the occurrence of self-description which is considered to have positive effects (cf. quote 16). Luhmann suggests two ways which will both be taken up in the further course of the thesis: First, he suggests that there should be special roles that are responsible for observing the flow of communications within the social system. Second, he suggests the technical means of written artifacts.

2.4.4 Self-descriptions as Guidance

Sd3: Self-descriptions are referred to by other communications and guide their flow.

So far it has been said that self-descriptions are communications that are part of the communicative network by which a social systems reproduces itself, but that they are distinct from other types of communications. The next question to be discussed is how self-descriptions and other acts of communication are related to each other.

Within Luhmann's theoretical framework, organizations are networks of communications; and communications were described as a three-part unity in section 2.3.3. These aspects are important but not sufficient for describing an organization as a unit: as long as communication successfully follows communication, there is a communicative network which may spread endlessly. To appear as a system that is distinguishable from its environment, an organization needs a boundary; such a boundary distinguishes between communications that are part of the organization's network and those that are not.

For any given communication within an organization there is a wide variety of possible connecting communications; each member of the organization has to decide which information he selects from the previous communication to start the next (cf. section 2.3.3). An organization's self-description reduces the number of possible communications by encoding rules and expectations (cf. section 2.3.6). And thereby self-descriptions become the means by which an organization creates and maintains its boundary.

Consider the following example (referring again to the scientific team and their efforts related to literature):

a) Wednesday evening one of the researchers skims through the tables of contents of all scientific periodicals available and places orders for three articles. He does so because the group decided on its last meeting, that orders should be placed by Thursday each week.

b) The student worker from the library needs a new PIN for her library account; she does not know who to call, so she checks the library's internet page to find out who is responsible for the administration of PIN's. She then calls the person to find out how to proceed.

Example a) describes how a researcher remembers an agreement and how he acts accordingly. In example b) the student worker refers to a written and published self-description in order to know who to ask for help.

An organization's self-description does not determine the flow of communication in the same way a computer program determines the flow of instructions executed by a processor. Instead self-descriptions are used by the members of an organization to help them select an „appropriate" communication. Luhmann uses different terms to explain the way self-descriptions are related to other communications: *„orientation"* (quote 1 in Table 1), *„steer"* (quote 6 in Table 1), *„help"* (quote 8 in Table 1), *„information", „guidance"* (quote 11 in Table 1).

An important implication is that communications and written artifacts are only self-descriptions if they are referred to by other communications. An organizational handbook that has been written, securely stowed away in the cupboard and forgotten, is no self-description because it is not used.

Figure 4 illustrates the influence of self-descriptions by several relations. First there is the relation *refers to* between *Reproduction of the Organization* and *Distinguishing Operations*; second the relation from the entity *Written, self-descriptive artifacts* into the process *Communicate* illustrates that communication within an organization relies on written artifacts; third, the zigzag arrow looping out from and back into *Communicate* illustrates how the process of communication remakes itself (cf. 2.3.2, p. 38 for a discussion on autopoiesis) and self-descriptions have an important part in this loop.

So self-descriptions are communicative acts and written artifacts that are used by members of an organization as guidance for their actions. On the social level of the organization, self-descriptions ensure that the organization does not fade out into an arbitrary network of communication but that communications that belong into the organization can be distinguished from those that do not.

For CSCW-projects the challenge will lie in the creation of self-descriptions that serve as guidance in the context of using a CSCW-application.

2.4.5 Subjects of Self-descriptions

Sd4: The subject of self-descriptions is the boundary between the social system and its environment.

So far it is established that self-descriptions are distinguishing communications, which are distinct from other communications in the organizations, but which are referred to by these other communications. The last aspect discussed to transfer the concept of self-description from the abstract level of theory to a more practical level, is the content of self-descriptions.

The *topic* of self-description is the *boundary of system and environment* (illustrated as an entity in Figure 4); when members of an organization talk about the rules and expectations within the organization, they perform self-descriptions. Luhmann mentions two specifications of this broad topic: First *„a connection between actions*" (quote 12 in Table 1), i.e. self-descriptions describe which actions can be expected as connecting actions to a previous one. Second, *„difference between compliance and deviation"*; i.e. self-descriptions describe the difference between communications that are considered acceptable and those that are not.

Consider the following examples:

a) A researcher sends an e-mail to the student worker that consists only of a list of 10 articles. The student worker understands that she is supposed to get these articles for the researcher.

b) Among other orders one researcher asks the student worker in the library to copy an article from a motorcycle magazine, because she considers buying a new motorcycle.

In example a) the student worker knows what is expected of him in reaction to the researchers e-mail; this knowledge was created by means of self-description. Either, somebody orally explained to him what the tasks of a student worker in the library are; or he even received a written job description. The communication in example b) could give rise to (self-descriptive) discussions within the organization, because a motorcycle magazine is not a typical scientific periodical and it is questionable whether a student worker should be responsible for copying such an article. Within such a discussion, rules and expectations existing in the group would be referenced to decide whether or not the researcher's behavior was acceptable or not.

2.4.6 Definition of Self-Description

The previous sections are summarized by my definition of self-descriptions that shall be used within this thesis.

Self-descriptions are communications by which an organization maintains its boundaries. The following characteristics apply to self-descriptions:

1. Self-descriptions describe which communications (actions) are acceptable within an organization and which are not.
2. Self-descriptions are referred to by other communications (actions) for guidance.
3. Self-descriptions take the form of both ephemeral communications and durable artefacts such as documents.
4. Self-descriptions can and should be distinguished from "normal" communications, because this increases the organization's chances for developing more demanding and specialized structures.

Term 11: Self-Description

The term „self-description“ is used for referencing documents as well as communicative processes by which such documents may be created. The terms self-descriptive artifact or self-descriptive document are used if the difference between a self-descriptive action and its results needs to be emphasized.

2.4.7 Self-Descriptions and Organizational Change

Quote 24 includes an interesting remark about the role of self-descriptions in processes of organizational change: by explicating its rules for appropriate communication, the organization creates the basis for a reflection that may lead to change. This implies that self-descriptions have a dual role in organizations: on the one hand they provide stability by serving as an orientation for the members [Willke (1994), p. 149]; on the other hand they are a helpful point of reference for change processes [Wollnik (1998), p. 147].

Socio-technical self-descriptions that will be defined in the next section should also serve both processes: First, they should document a status quo which is then reflected and changed with respect to a new CSCW-system. Second, the new socio-technical self-description should guide the usage of the CSCW-system and thus stabilize the organization. The new socio-technical self-description can then be subject to discussion and change again, and thus a continuous process of improvement is supported.

A brief digression on the term organizational change and organization development shall be included at this point. The term organizational change refers to organizational change processes in general, whereas organization development refers to a deliberate change with preset goals (cf. [Beckhard (1969), p. 9], [Bennis (1969), p. 10], Gebert, Gawlik, Herzig (1974)]. A prominent definition of organization development is [French, Bell (1973), p. 15]:

> *"… organization development is a long-range effort to improve an organization's problem-solving and renewal processes, particularly through a more effective and collaborative management of organization culture – with special emphasis on the culture of formal work teams – with the assistance of a change agent, or catalyst, and the use of the theory and technology of applied behavioral science, including action research."*

Structural coupling between an organization and a CSCW-system is one form of organizational change which may occur as an organization development project or may happen without any planning in form of accidental metamorphosis. The term organizational change is the broader concept and therefore I use it within this thesis.

The term **organizational change** is used in this thesis to refer to change processes affecting an organization in general. Organization development is one form of organizational change that is characterized by a planned approach and a preset goal.

Structural coupling (cf. Term 10) between an organization and a CSCW-system is regarded as a specific type of organizational change.

Term 12: Organizational Change

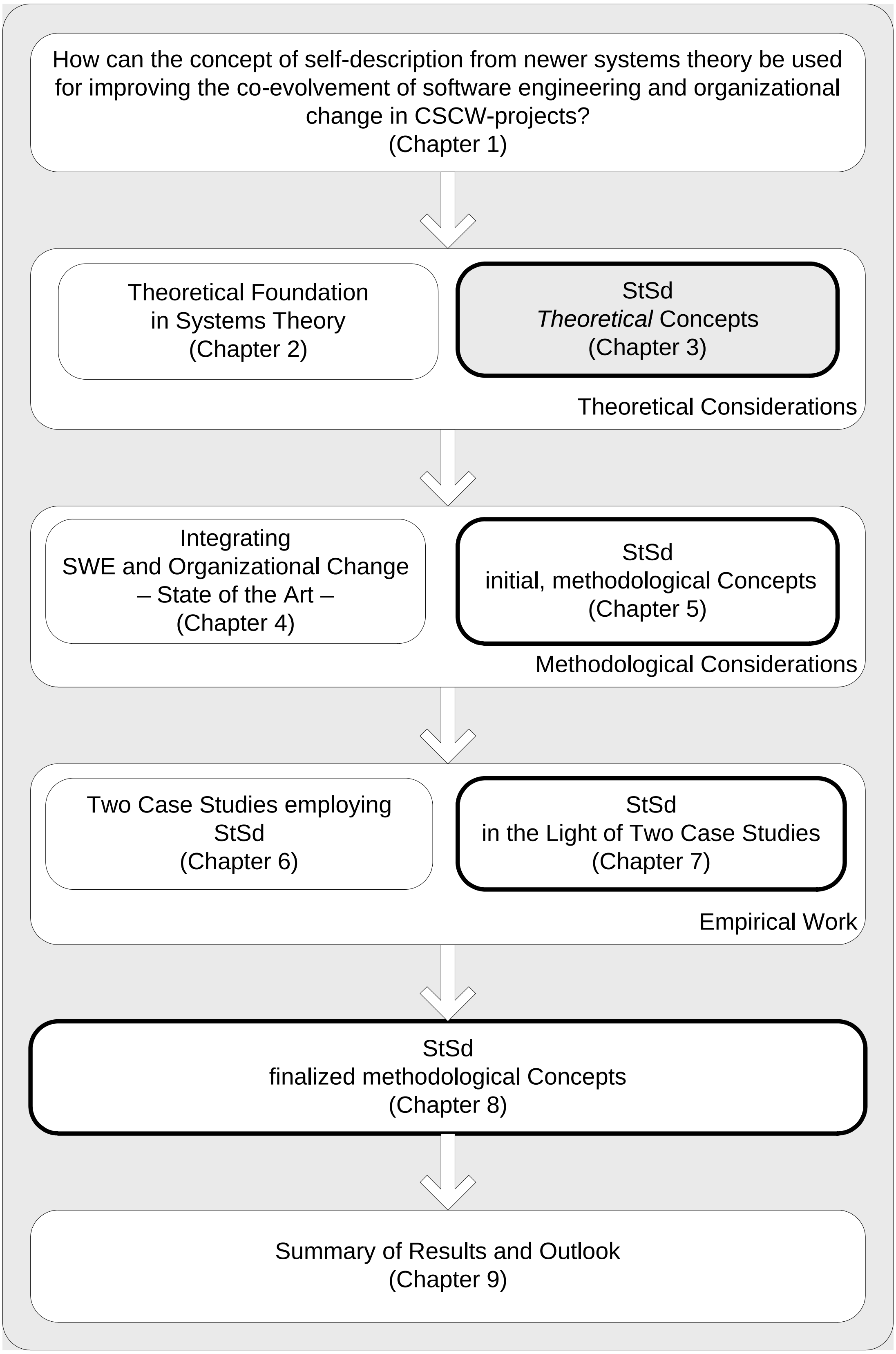
2.5 Summary

In the previous sections Luhmann's concept of organizations as autopoietic systems consisting of a network of communications was introduced. This concept provides explanatory potential for many research areas in computer science. Obvious links exist to the field of multiagent systems: it is the special characteristic of multiagent systems – in contrast to merely distributed systems – to view agents as being autonomous. Agents can be influenced by changes in their environment, but it is not possible for agents to determine each other's behavior, e.g. by directly invoking methods [Wooldridge (2002), pp. 163].

But my goal in this thesis is to use these theoretical concepts for a reconceptualization of „socio-technical" systems that is helpful in designing methodological support for CSCW-projects. Table 2 summarizes the main results and their implications, and the following chapter 3 contains the transfer from “social and technical” to “socio-technical”.

Concepts of newer system theory	**Implications for the conceptualization of ”socio-technical system”**	**Implications for methods supporting CSCW-projects**
Technical systems are allopoietic while social systems are autopoietic.	A theoretical construct „socio-technical system” needs to make these differences visible for two reasons: first it should be able to describe and explain the differences between technical and social systems, second it should be usable as a basis for the elaboration of methods supporting CSCW-projects.	CSCW-projects address two different types of systems and need to include different sets of methods appropriate for the respective system.
Technical systems consist of firmly coupled operations while social systems have to cope with the phenomenon of double contingency.		While engineering methods are appropriate for the technical elements in a CSCW-project, methods handling the contingencies of human communication are needed for the organizational elements.
Social systems react to relevant changes in their environment by structural coupling.	A theoretical concept of „socio-technical system” should include the relations between a social and a technical system such that they can be subject to intervention.	Adapting work procedures such that they correspond to a new CSCW-system can be described as a process of structural coupling. There should be methods available for CSCW-projects that support the structural coupling of an organization with respect to the CSCW-system.
Social systems need self-descriptions to define their boundaries and thereby their identity.	A theoretical concept „socio-technical system” should describe how an organization’s self-descriptions changes when it becomes „socio-technical”.	Structural coupling with respect to a CSCW-system implies that the CSCW-system needs to be included into the organization’s self-description. Thus, methods supporting an organization in modifying and maintaining its self-description to include references to the CSCW-system should be included in CSCW-projects.

Table 2: Systemic Concepts - Summary and Implications


How can the concept of self-description from newer systems theory be used for improving the co-evolvement of software engineering and organizational change in CSCW-projects?
(Chapter 1)
Theoretical Foundation in Systems Theory
(Chapter 2)
StSd
Theoretical Concepts
(Chapter 3)
Theoretical Considerations
Integrating SWE and Organizational Change – State of the Art –
(Chapter 4)
StSd
initial, methodological Concepts
(Chapter 5)
Methodological Considerations
Two Case Studies employing StSd
(Chapter 6)
StSd
in the Light of Two Case Studies
(Chapter 7)
Empirical Work
StSd
finalized methodological Concepts
(Chapter 8)
Summary of Results and Outlook
(Chapter 9)

3 Socio-Technical Self-Descriptions (StSd) – Theoretical Concepts

In this chapter I extend Luhmann's concepts by explicitly including CSCW-systems into the terms and models. This is done in two steps: First, section 3.1 takes Figure 4 and answers the following question: What happens to the description if a CSCW-system is added to the environment of the organization? The multi-facetted relations between an organization and a CSCW-system are unfolded; the central concept of self-description (cf. Term 11) is extended to become "socio-technical" and discussed in relation to important terms from the field of CSCW such as groupware conventions, articulation work, appropriation, or boundary objects. In a second step, section 3.2 derives a set of definitions which constitute the reconceptualization of "socio-technical" systems, the necessity of which was argued in section 2.1.5. Section 3.2.2 concludes by discussing implications for CSCW-projects that provide a bridge to the discussion and elaboration of methodological support.

3.1 Self-descriptions in the Context of CSCW-Systems

3.1.1 From Social to Socio-Technical

The first idea for extending Luhmann's concepts of social systems and self-description to a concept of „socio-technical systems" is simple: take the graphic in Figure 4 and add a CSCW-system in the environment. Figure 5 shows the result of such a first, simplified transfer. A relation between *communicate* and the *CSCW-system* is added (1) to indicate that the CSCW-system is used by the communications (actions) within the organization. A logical consequence which is already included in the graphic of Figure 4 is that the organization's *self-description* now includes information about the usage of the *CSCW-system*. Because this is important, it is illustrated by an extra relation (2) between *Boundary of System / Environment* and the usage of the *CSCW-system.*

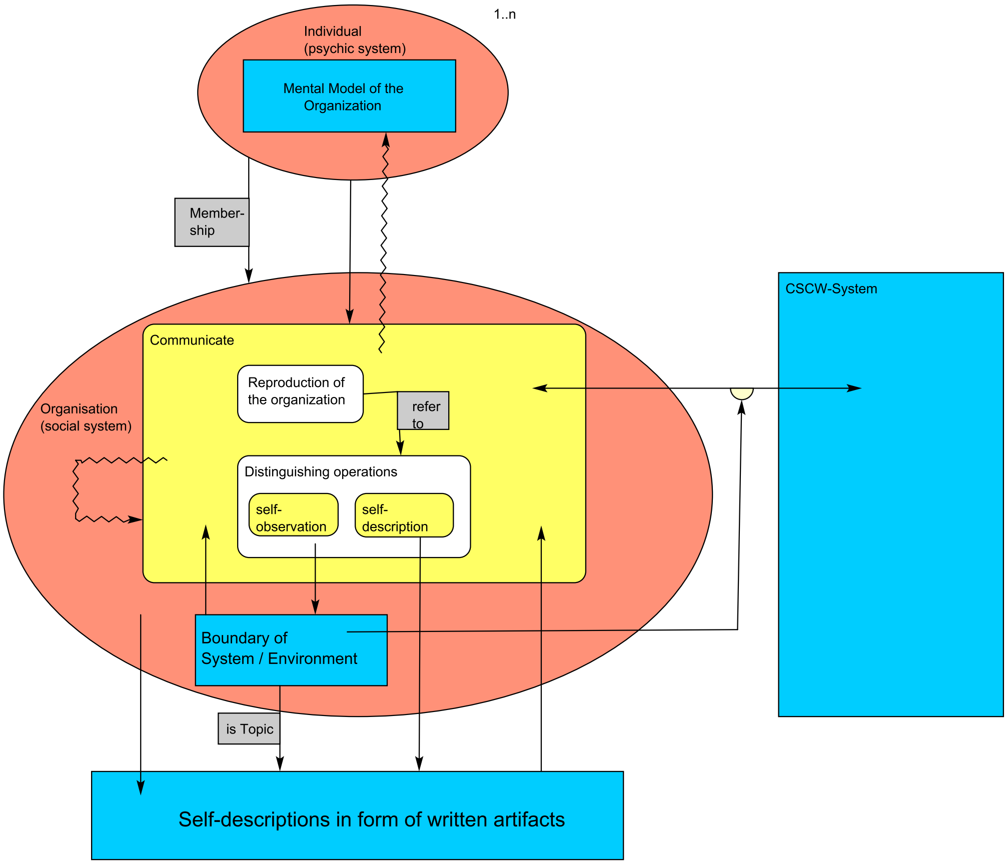

1..n
Individual
(psychic system)
Mental Model of the Organization
Member-ship
CSCW-System
Communicate
Reproduction of the organization
refer to
Organisation
(social system)
Distinguishing operations
self-observation
self-description
Boundary of System / Environment
is Topic
Self-descriptions in form of written artifacts

Figure 5: Socio-Technical Setting (simplified)

The complex illustrated by Figure 5 is called a „socio-technical setting". The socio-technical setting comprises both, the social and the technical system. The term „socio-technical setting" is used instead of „socio-technical system" for keeping it apart from conceptualizations like the ones cited in section 2.1.3. At the end of this chapter (cf. section 3.2.1) definitions for the terms socio-technical setting and socio-technical system are given based on the discussions in the following sections.

Because the *Boundary of System / Environment* is a topic in the organization's self-descriptive artifacts, there should now be documents containing information about the usage of the CSCW-system within the organization. The creation and maintenance of such documentation is indeed central to the methods I elaborate; however the conceptualization of Figure 5 is far too simplified. The relation between an organization and a CSCW-system has been subject to many studies; and many results show that the relation is much too complex to be reduced to „usage". Examples for topics that bring in additional complexity are:

- Durable forms of self-descriptions are not limited to *written artifacts* as implied by Figure 4. Technical artifacts are self-descriptive because they contain inscribed social rules (cf. section 2.2.2).
- The CSCW-system itself contains self-descriptive aspects such as predefined workflows or access rights.

- If one takes a closer look at the *self-descriptive artifacts* the topic of coordinative artifacts which has been studied extensively in the field of CSCW is not far.
- Figure 5 implies that the *CSCW-system* is used in a communicative context only. This contradicts the fact that a CSCW-system can very well be used for individual purposes; and that phases of individual use alter with phases of communicative use.

Therefore a more detailed view needs to be taken on the relation between an organization and a CSCW-system. Figure 6 illustrates the result of such a view and the following sections discuss the details referring to Luhmann's system theory as well as to relevant results of CSCW-related research.

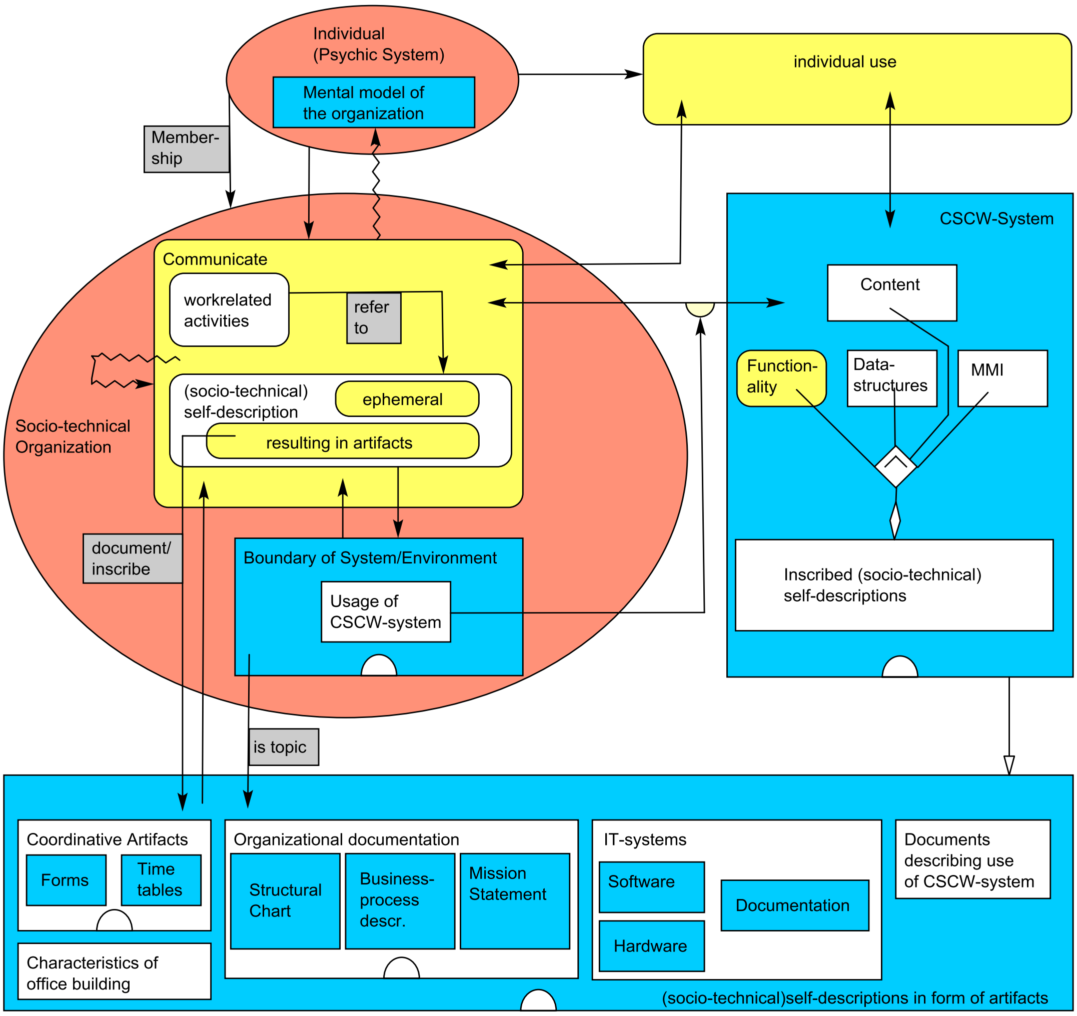
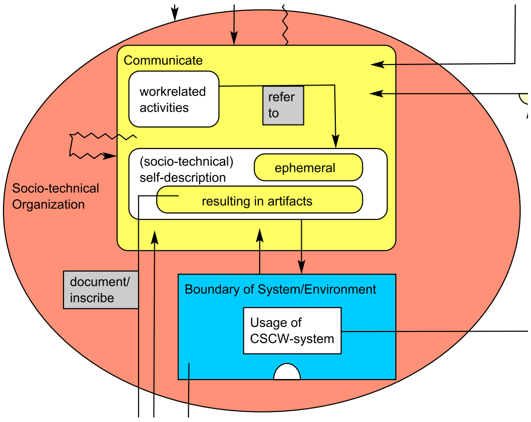


Figure 6: Socio-technical Setting

Figure 6 shows the interdependencies between a CSCW-system and an organization in which it is used. The following subsections explain and discuss the elements of Figure 6, such that section 3.2.1 can conclude by providing definitions for socio-technical setting, socio-technical organizations and socio-technical self-descriptions.

For the ease of reading, details of Figure 6 are repeated in the subsections.

3.1.2 Self-Descriptive Communication



Workrelated Activities vs. Self-Description

The process *Communicate* that is already known from Figure 4 has been modified for Figure 6. It now contains the subactivities *workrelated activities* and *(socio-technical) self-descriptions*. Although the terms *distinguishing operations* and *Reproduction of Organization* are more precise with regard to Luhmann's theoretical concepts, they are not intuitive for an audience concerned with CSCW-projects. Therefore Luhmann's own idea of describing communication by actions (cf. section 2.3.4) is used and *workrelated activities* represents those communications by which the organization keeps itself alive. *Self-descriptions* stand as a pars-pro-toto for the *distinguishing operations* (cf. section 2.4, p. 53) and denote those communications by which the organization describes its own actions.

So, within an organization two kinds of communications are distinguished: the communications that are needed to get the work done and the communications needed to describe the organization. The former are called *workrelated activities* and the latter *(socio-technical) self-description.*

(Socio-technical) Self-description is further detailed into *ephemeral* and *resulting in artifacts*. This distinction coincides roughly with Luhmann's distinction between *self-observation* and *self-description* (cf. Figure 4). *Ephemeral* self-descriptions are communications on a meta-level that are used to coordinate other communications; these communications however do not result in artifacts, they are volatile. Self-descriptions *resulting in artifacts* are those self-descriptions that lead to documents or other artifacts that render the self-description more durable. The artifacts that contain parts of an organization's self-description are discussed in detail in section 3.1.3. The next section discusses how the deployment of a CSCW-system changes the topics of self-description.

Socio-technical Self-descriptions: Usage of CSCW-system becomes a Topic

The fact that self-descriptions are about the distinction between communication that belongs to the organization and communication that does not is discussed in section 2.4. When an organization uses a CSCW-system, the usage itself will become topic of the self-descriptions. This idea is central to the methods I elaborate: an organization can only adopt a CSCW-system if it includes it into its self-descriptions. The organization needs to develop rules or conventions for using the CSCW-system such that other activities can refer to them for guidance, if not, the usage of the CSCW-system will be uncoordinated and less effective. Self-descriptions that include information about the usage of a CSCW-system shall be called socio-technical self-descriptions. Socio-technical self-descriptions share the basic characteristics of social self-descriptions (cf. Term 11, p. 57) but extend them with respect to the CSCW-system. At this point a preliminary definition of socio-technical self-descriptions is given; a more detailed definition including the results of the discussions presented in the next sections is given in section 3.2.1 (Term 21)

A **socio-technical self-description** is a self-description that includes information about the usage of a CSCW-system.

Term 13: Socio-Technical Self-Description (preliminary Definition)

To include the CSCW-system and its usage into an organization’s self-descriptions means to communicate about the CSCW-system. The members of the organization need to discuss and negotiate the way in which they want to use the CSCW-system. By doing so, the usage of the CSCW-system is raised from the individual to the social level. Training courses in the context of software engineering projects are often limited to teaching the individuals how to handle an application. Such courses fall short with respect to CSCW-systems because they do not address the social level. For CSCW-projects it is also important to support the organization in communicating about the usage of the CSCW-system and thereby including it into its self-description.

The activity *(Socio-technical) self-description* contains parenthesis around „socio-technical” to indicate that socio-technical self-descriptions do not completely replace „normal” self-descriptions. Self-descriptions such as organizational charts or mission statements might not be changed at all by the deployment of a CSCW-system; in an organization, in which a CSCW-system is used, there will be self-descriptions that make no reference to the CSCW-system and there will be socio-technical self-descriptions that describe how the CSCW-system is adopted. The degree to which self-descriptions turn into socio-technical self-descriptions, i.e. the number of references made by various self-descriptive artifacts to a CSCW-system, could be a measure for the degree of structural coupling. The following section discusses this idea.

Socio-technical Self-descriptions as a Means and a Measure

There are two different perspectives under which socio-technical self-descriptions can be regarded: first they are a means for supporting structural coupling, second they can become a measure for the degree of structural coupling that has taken place.

The extension of self-descriptions to socio-technical self-descriptions can be described as a field for intervention and methodical support. The assumption is that the explicit discussion and documentation of how the CSCW-system should be used within the cooperative work procedures is advantageous for its adoption. This assumption is motivated by Luhmann's hypothesis that those organizations that allow distinct self-descriptive communications are more likely to develop complex structures (cf. Table 1, quote 15) than those that do not. A consequence for CSCW-projects is that they should include methodological support for discussing and documenting computer supported work procedures, because this helps the organization in planning and realizing more complex scenarios of cooperative use. This idea is central to the framework of socio-technical self-descriptions which includes a format for documenting socio-technical self-descriptions (cf. section 5.2.2) as well as a workshop design for enabling a communicative discourse (cf. section 5.2.3).

The second perspective is to interpret the inclusion of *Usage of CSCW-system* in the organization's self-description as a measure for the degree of structural coupling. If it is advantageous for an organization to include the *Usage of CSCW-system* into its self-description, then one could turn the argumentation around: the more traces of the CSCW-system can be found in the organization's self-descriptions the more the organization has adopted the CSCW-system. The following examples taken from Figure 6 give an idea how references to a CSCW-system can be included into self-descriptive artifacts:

- The structural chart of the organization includes functional roles for maintaining the CSCW-system.
- Business process descriptions include information about the use of the CSCW-system.
- Paper forms include references to information found in the CSCW-system.
- The layout of the common rooms in an office building encourages collaborative use of the CSCW-system i.e. by offering workstations.

The aspect of measuring the degree of structural coupling does not lie in the focus of this thesis, because the concept of socio-technical self-descriptions needs to be elaborated first. But if, at the end, there is a concept of socio-technical self-descriptions, one could think of employing it as a measure.

Two concepts discussed in the field of CSCW are linked to the concept of socio-technical self-descriptions as it has been described so far: articulation work and groupware conventions. They shall be discussed in the following two sections.

Socio-technical Self-description and Articulation Work

Articulation work is a concept which is central to CSCW-related research, at least since Schmidt's and Bannon's important article „Taking CSCW seriously. Supporting articulation work." [(1992)]. The term itself has been described earlier for the field of office information systems by Gerson and Star [(1986), p. 266]. Schmidt and Bannon argue that since cooperative work is by definition distributed (cf. Term 2), extra effort is needed for coordinating the parts. This extra effort is referred to as articulation work [Schmidt, Bannon (1992), p. 18]:

> *„... the distributed nature of the arrangement must be kept in check, managed. The distributed activities must be articulated. Articulation work arises as an integral part of cooperative work as a set of activities required to manage the distributed nature of cooperative work."*

Cooperative work comprises interdependent activities carried out by different people; articulation work is necessary to coordinate these activities. *„The word ‚articulate' is used in the sense of „to put together by joints"* [Schmidt, Simone (1996), p. 158]. Articulation work needs artifacts for coordinating the interdependent activities. Schmidt and Simone describe this need as follows [ibid. p. 162]:

> *„In cooperative work settings characterized by complex interdependencies, the articulation of the distributed activities requires special artifacts which, in the context of a set of conventions and procedures, are instrumental in reducing the complexity of articulation work and in alleviating the need for ad hoc deliberation and negotiation."*

Examples of such coordinative artifacts are checklists or timetables. Artifacts will be discussed in more detail in section 3.1.3; the question discussed first is: What is the relation between self-description and articulation work? The answer is twofold. First self-description is a prerequisite for articulation work, because self-descriptive communications provide the elements (conventions, procedures, artifacts) that are needed for successful and effective articulation work. Self-description refers to those communications that describe the organization of work in general on an abstract level; articulation work on the other hand comprises concrete activities needed everyday in order to successfully coordinate cooperative work.

Consider an example referring the group of scientists and their scientific periodicals: once the person responsible has imported new tables of contents into the system, she sends a mail to the whole group informing everybody, so that they know that they can proceed with their work. The act of sending the e-mail is articulation work, because it is needed to ensure that the cooperative reading of scientific periodicals functions. Sending the e-mail however is no self-description, it is a „normal" *workrelated activity* (cp. Figure 6). The self-descriptive communication took place earlier when the group discussed and decided that sending e-mails is the means to inform everybody that new tables of contents are available.

From a second point of view however, there is an overlap between the concepts, because it is hard to imagine that articulation work functions without self-descriptive elements. This is true especially in those cases, where *„ad hoc deliberation and negotiation"* are needed. In these cases it is very likely that articulation work includes communication about the *workrelated, cooperative activities* which would be classified as *ephemeral self-description* in the graphic of Figure 6.

In his work on articulation work Schmidt focuses on the goal that CSCW-systems need to support articulation work. For this purpose it is important to show how articulation work is interwoven with „normal" work with the goal of designing technical features that support both. With regard to self-descriptions it has been stated that it is advantageous for an organization to clearly distinct between self-descriptive and other communication (cf. section 2.4.1). So the purpose of the theoretical and methodological concepts of this thesis is to describe self-descriptions as a form of communication that is distinct from normal work but that provides support for it. Articulation work is needed to coordinate the distributed parts of cooperative work; therefore CSCW-systems need to support articulation work. But then, the usage of CSCW-systems itself needs articulation and I elaborate the concept of socio-technical self-descriptions as a means to support this need.

Socio-technical Self-descriptions and Groupware-Conventions

The concept of (socio-technical) self-descriptions is closely related to the topic of conventions. Conventions are one type of self-descriptions: they describe the activities or more general the behavior that members of an organization can expect from each other. Mark cites Lewis and provides the following *„definition of normative convention behavior in physically collocated work"* which is adopted for this thesis.

> "*A regularity* R *in the behavior of members of a population* P *when they are agents in a recurrent situation* S *is a convention if and only if it is true that, and it is common knowledge in* P *that, in any instance of* S *among members of* P,
>
> - *[Almost] everyone conforms to* R*;*
> - *[Almost] everyone expects everyone else to conform to* R*;*
> - *[Almost] everyone prefers to conform to* R *on condition that the others do, since* S *is a coordination problem and uniform conformity to* R *is a coordination equilibrium in* S*."*
>
> Lewis (1969) cited in [Mark (2002), p. 356]
>
> *"Explicit conventions can take the form of agreements, rules or even laws …"*
>
> [ibid. p. 363]

Term 14: Convention[17]

The evolvement of groupware-conventions was analyzed in the context of the POLITeam-system ([Prinz (1998)], [Mark (2002)]). The POLITeam-system is a groupware-system for supporting the work on shared documents in distributed administrative organization. The setting was given by the move of Germany's capitol from Bonn to Berlin during the 1990's. Mark and Prinz describe four phases of groupware usage developing from individual use to group orientation. At the beginning the users focused on the possibilities the system offered for their individual work, they were busy learning the basic functionalities. But once they grew comfortable with the individual use, they discovered the social dimension. Mark [(2002), p. 376] describes how conventions slowly evolved because no one felt competent to suggest and enforce them earlier in the project. Prinz suggests that this evolvement of groupware-use from individual to group-oriented may be typical; however long-term studies are missing to support or dismiss this hypothesis [Prinz (1998), p. 148].

There are two conclusions to be drawn from the POLITeam research on groupware conventions. First, additional conventions are necessary. Mark takes the empirical evidence to argue convincingly that successful groupware usage requires conventions [Mark (2002), p. 371 and p. 381]. This insight is confirmed by both of my case studies during which the participants documented rules for the use of CSCW-systems as part of their socio-technical self-description (cf. section 7.1.1.4).

[17] The word [Almost] is not included in the orginial quote; Mark however argues that the definition in this form is still in line with Lewis' original intentions.

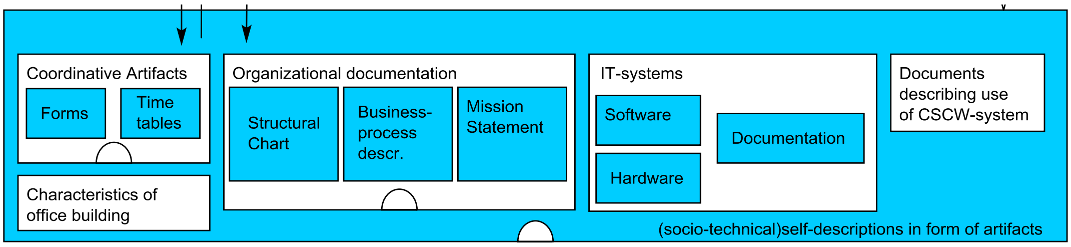
The second conclusion concerns the support for agreeing on groupware conventions. Shouldn't it be possible to support the adoption of the CSCW-system on the social level from the very beginning of a project? The methods I elaborate and evaluate attempt to do so. A very illustrative example is the design of the training courses offered in both case studies: they consisted of two parts (cf. Example 34). In the first part, the functionalities of the system were explained and trained. This corresponds to the phase of individual learning. In the second part which immediately followed the first one, the groups discussed the necessity of further „agreements" that were needed to successfully use the CSCW-system. During this second part the participants also negotiated and documented some conventions with which they would start using the system.

This approach of early communication about the usage of the CSCW-system is also encouraged by findings regarding „technological frames" [Orlikowski, Gash (1994)]. The authors describe how individuals bring in different backgrounds (i.e. frames) for the utilization of IT-systems. Aspects included are: *„assumptions, knowledge, and expectations"* [ibid, p. 176]. Interpreting a case study on the deployment of Lotus Notes®, the authors show how diverging technological frames lead to a usage of the system that remained way behind what was intended by the management when investing in it. In particular, most users did not think of Lotus Notes® as a system for supporting cooperation, their intentions were oriented toward an individual use. To cope with the problem Orlikowski and Gash suggest that [ibid, p. 202]:

> *„Early articulation, reflection, discussion, negotiation, and possibly change of inconsistencies and incongruences may reduce the likelihood of unintended misunderstandings and delusions around the implementation and use of a new information technology."*

This is what socio-technical self-descriptions may do in early phases of CSCW-projects: they may help developing congruent technological frames among the participants and future users.

3.1.3 (Socio-technical) Self-descriptions in Form of Artifacts


Coordinative Artifacts
Forms
Time tables
Characteristics of office building
Organizational documentation
Structural Chart
Business-process descr.
Mission Statement
IT-systems
Software
Hardware
Documentation
Documents describing use of CSCW-system
(socio-technical)self-descriptions in form of artifacts

Documents describing use of CSCW-system

Section 2.4.1 discusses that self-descriptions take the form of ephemeral communication as well as artifacts. The latter are important because they render the self-descriptions more durable and make it easier for other communications to refer to it. Socio-technical self-descriptions are defined above as self-descriptions that include references to a

CSCW-system. The artifacts resulting from socio-technical self-descriptions must describe organizational as well as technical issues, because they have to document the usage of a CSCW-system (cp. section 3.1.2, p. 64).

This thesis suggests creating dedicated documentation containing an organization's socio-technical self-description (cf. section 5.1); the entity *Documents describing use of CSCW-system* illustrates this form of self-description. But it would be wrong to assume that these dedicated documents are the only form of socio-technical self-description. Other forms exist and emerge during a CSCW-project. The most important alternate form are self-descriptions that are inscribed into the software of the CSCW-system itself. There are interdependencies between these two that can be described on a theoretical level (cf. section 5.1.2) and that bear implications for the form and content of socio-technical self-descriptions in the course of CSCW-projects (cf. section 7.2.2). Case study 1 will also provide an example where different forms of (socio-technical) self-descriptions were seen to be competing with each other (cf. Example 26, p. 210).

The following sections discuss examples from Figure 6 and CSCW-research related to artifacts.

Extending the Concept of „written artifacts"

In Figure 4 self-descriptive communications lead to *self-descriptions in form of written artifacts*. In Figure 6 this entity is replaced by a more general entity labeled *(socio-technical) self-descriptions in form of artifacts* which contains a selection of examples what such artifacts could be. This has been done to illustrate the wide variety of artifacts that can contain self-descriptive aspects; they are by no means limited to written documents in the narrow sense of the term. Luhmann stated that self-descriptions do not need to take the form of *„canonical documents"* i.e. they may be distributed over various documents. But then, what are documents? Discussions like the ones summarized in [Buckland (1997)] show that limiting documents to written texts is much too narrow. So, *(socio-technical) self-descriptions in form of artifacts* shall include more than written documents. Consider the following examples included in Figure 6:

- The characteristics of a building can carry information about an organization's self-descriptions. If – for example – all offices have glass windows into the hallway, this could indicate that everybody may know who is present and who talks to whom.
- Coordinative artifacts like forms or timetables [Schmidt, Simone (1996)] are the results of self-descriptions because they include information about the ways in which people organize their cooperative work.
- IT-systems in general and CSCW-systems in particular contain parts of an organization's self-descriptions e.g. in form of access control rules mirroring hierarchical structures or hard coded action sequences mirroring business processes.
- Organizational documents such as structure charts, mission statements or business processes are a traditional way of documenting parts of an organization's self-description.

The following sections discuss the relation between socio-technical self-descriptions and artifacts in CSCW in more detail.

Socio-technical Self-descriptions and Coordinative Artifacts

Much CSCW-related research is done to analyze how people use artifacts to coordinate their cooperative work (e.g. [Schmidt, Simone (1996)], [Robinson (1993a)]). The concept of articulation work has been discussed above, and it has been mentioned that articulation work relies on *„special artifacts"* [Schmidt, Simone (1996), p. 162]. This section shall discuss the role of artifacts and their relation to the concept of self-description.

Cooperative work needs artifacts because they help to coordinate the distributed parts of work. Schmidt and Simone put it [(1996), p. 176]:

> *„The role of the artifact in a coordination mechanism is fundamentally to objectify and give permanence to the coordinative protocol so that its stipulations are unceasingly publicly accessible."*

The coordinative protocol that is mentioned in the citation refers to plans, conventions or procedures that are used together with the artifact (cf. section 3.1.2, p. 68). The goal of understanding articulation work and coordinative artifacts is, to be able to design CSCW-systems that do not accidentally destroy complex coordination mechanisms by replacing artifacts with inadequate computer systems. The design of CSCW-systems often focuses on sequentializing actions. Process models of workflows are created and hard coded; business process reengineering (BPR) is the most prominent example of this method. CSCW-researchers argue that by doing so, the complex mechanisms of coordinating work in the unpredictable everyday situations are not supported anymore (e.g. [Suchman (1987)]). One reason is that the complex coordination mechanisms established around coordinative artifacts are not transferred to the CSCW-system. Robinson ([(1993b)],[(1993a)]) suggests grounding the design of CSCW-systems not on sequences of action but on common artifacts which are modified by the actors as needed. His very illustrative example is that of a hotel's key rack. Such a key rack is much more than a place to hang keys. Together with the convention that guests should deposit their keys at the reception when leaving the hotel[18], the keys on the rack indicate which rooms are currently empty. Key racks usually have „pigeonholes" for each room that can be used to pass messages from the hotel management to the guests or even among the guests; and at night when all guests are in their rooms, the key rack gives an impression of the number of vacancies in the hotel. Such a passive, physical artifact like a key rack can be the bearer of information relevant for cooperative work and thereby influence the course of action. Another theoretical school, behavior setting, also emphasizes the influence physical elements in the environment have on human behavior [Pankoke-Babatz (2003)].

What is the relation between such artifacts and self-description? Again, the answer is twofold. First, self-descriptions are a prerequisite for designing common artifacts. Consider the example of the hotel's key rack again. The hotel management has to discuss and decide whether information about the guests' presence should be public or

[18] The enforcement of the convention that guests should not take their keys with them when leaving the hotel is one of Latour's well known examples for the inscription of social conventions into physical artifacts (cf. chapter 1, p. 17); but although the examples are so close, Robinson does not refer his considerations to Latour's actor-network theory.

not. This discussion results in a self-description of the hotel as an organization. Depending on the decision the key rack is either put up such that it is visible to everybody at the reception or it is hidden from view e.g. by putting it under the desk in a drawer-like construction. This simple example shows how physical artifacts can carry elements of an organization's self-description, and leads to the second aspect of the answer. Coordinative artifacts bear parts of the organization's self-description and once they have been implemented, the self-description is firmly manifested. Section 5.1.2 discusses how the different manifestations of socio-technical self-descriptions are related within a CSCW-project.

At this point however, an important difference between coordinative artifacts and (socio-technical) self-descriptions needs to be emphasized. It has been discussed in section 2.4 that self-descriptions take place on meta-level, communicating about communications. Self-descriptions in this sense are long lasting and ensure the identity of the organization. The example of the key rack includes yet another form of self-description, a more situational one: the state of the key rack (i.e. number of keys hanging on the rack; number of notes in the pigeonholes) provides information about the current state of the hotel and its inhabitants. This is also a kind of self-description – a part of the hotel (the key rack) describes the state of the hotel. This form of self-description does not take place on a meta-level, it is situated; it is not long lasting but rather changing as the normal daily procedures take place in the hotel; it does not purport identity but rather signals changing states. This latter form of self-description is closer to the concepts of coordinative artifacts and articulation work; however it is not in the focus of this thesis. In order to keep the two forms of self-descriptions apart, the term „momentary self-description" is introduced:

> The term **momentary self-description** shall be used for information generated by a system describing its own current state. Momentary self-descriptions can be generated by technical as well as social systems.

Term 15: Momentary Self-Description

Socio-Technical Self-Descriptions as Boundary Objects

Another concept that needs to be discussed in the context of artifacts is that of boundary objects, which was introduced by Star in 1989. Boundary objects are objects that can be used by heterogeneous groups that share some common goal or interest. People working in heterogeneous groups have the problem that although they may work on the same problem, they do not necessarily have a good model of each other's work, they employ different units of analysis or different abstraction of data, they may have different goals and time horizons etc. [Star (1989), p. 46]. But nevertheless, heterogeneous groups do succeed in cooperating and boundary objects support them [ibid.]:

> *„Boundary objects are objects that are both plastic enough to adapt to local needs and constraints of the several parties employing them, yet robust enough to maintain a common identity across sites. They are weak structured in common use, and become strongly*

> *structured in individual-site use. Like the blackboard, a boundary object „sits in the middle" of a group of actors with divergent viewpoints."*

Star derives several types of boundary objects including repositories, ideal types, terrains with coincident boundaries and forms and labels. Typical, often cited, examples of boundary objects are forms into which bird watchers write their observations. Biologists rely on amateurs for watching and counting birds; obviously a professional biologist could enter different, maybe more detailed, data into an observation sheet than an amateur. However the form can be designed in such a way that an amateur can enter his observations as well as the professional biologist. Such a form is a boundary object that adapts to the needs of amateur bird watchers as well as professional biologists; it supports both groups in their joint effort of watching and registering birds.

What is the relation between self-descriptions and boundary objects? At first sight self-descriptions are no boundary objects, because they are not meant to cross boundaries between different groups. On the contrary: they are meant to define and establish an organization's boundary. But at second sight there are usages within a CSCW-project where an organization's self-descriptive artifacts are used by different and even heterogeneous groups. Documents created in the course of requirements elicitation are an example: they contain parts of the organization's self-description such as work procedures and they are to be interpreted by software engineers who encode the self-description into software. Also in large organizations it is not possible to let all future users participate in all discussions, so for practical reasons it is necessary to pass socio-technical self-descriptions from one group of users or – more general – stakeholders to the other. And in this context socio-technical self-descriptions need to possess characteristics of boundary objects.

The fact that artifacts of socio-technical self-description will be used as boundary objects in the course of a CSCW-project will influence the selection of suitable forms of representation (cf. section 5.2.2). Especially case study 1 provides experiences in using diagrams representing socio-technical self-descriptions as boundary objects between different groups of stakeholders (cf. section 7.4.1). Both case studies show that the benefit of having socio-technical self-descriptions as boundary objects induces costs in form of additional effort in a CSCW-project; but they also show that the participants valued the outcome of the additional effort (cf. section 7.3.1, pp. 229).

Socio-technical Self-descriptions and Inscription

The fact that social „aspects" are transferred into physical artifacts (like the hotel key rack discussed above) and that these in turn influence social behavior again has been described from different viewpoints:

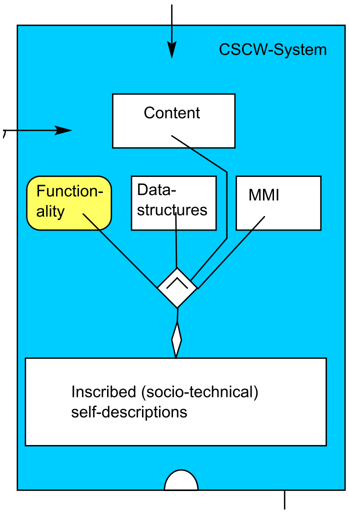
- Latour describes how conventions are transferred into technical artifacts and calls this process „inscription" (cf. chapter 1, p. 17).
- Activity theory describes the circle of externalization (process by which cultural elements are transferred to technical artifacts) and internalization (process by which technical artifacts are included into the course of action) (cf. chapter 1, p. 16).

- Behavior setting theory describes how the physical elements in the environment influence the behavior of people acting in the environment (an introduction into behavior setting theory and its implications for the design of CSCW-systems can be found in [Pankoke-Babatz (2003)]).
- For the field of CSCW, Simone and Schmidt elaborate the concept of artifactually imprinted protocols, which refers to conventions that are objectified within artifacts [Schmidt, Simone (1996), p. 167].

What is the relevance of inscription for socio-technical self-descriptions? CSCW-systems that contain inscribed social conventions contain self-descriptions. So far, the CSCW-system and its usage have been viewed as being an object of socio-technical self-descriptions. The organization needs to describe how its communications make use of the CSCW-system. Inscription now opens an additional point of view: the CSCW-system is also bearer of self-description; it is itself a self-descriptive artifact. The graphic in Figure 6 illustrates this relation in three ways: First, the entity *IT-system* is included in the larger entity *(Socio-technical) Self-descriptions in form of Artifacts* to indicate that IT-systems are bearers of self-descriptions. Second, the entity *CSCW-System* is connected to the entity *(Socio-technical) Self-descriptions in form of Artifacts* by a „is-a" relation indicating that the *CSCW-System* is not only used by the organization but also contains self-descriptive elements. Third, these self-descriptive elements are included in the entity *CSCW-System* as a subentity *Inscribed (socio-technical) self-descriptions*. For illustrating that *Inscribed (socio-technical) self-descriptions* are no additional element in a software-system, it is shown that they are made of *Functionality, Data-Structures, MMI,* and *Content* of the software[19].

Analyzing the relation between socio-technical self-description and CSCW-system yields three different aspects which are discussed in the next section.

(Socio-Technical) Self-Description and CSCW-systems



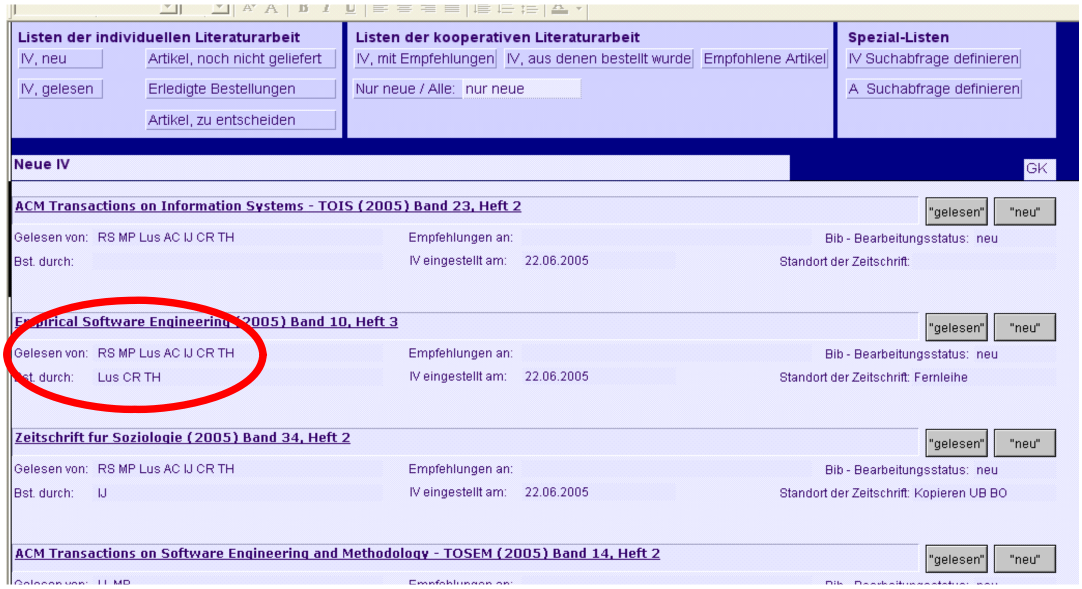
The relation between organizational self-descriptions and CSCW-systems can be described from three different points of view:

- Inscription: a CSCW-system is bearer of parts of the organization's self-description
- Content: a CSCW-system can be used for storing self-descriptive documents
- Socio-technical self-description: the coordinated use of a CSCW-system requires additional – socio-technical – self-description, in which the organization describes how the CSCW-system is to be included into its communicative network.

[19] This specification of *CSCW-system* follows a functional or object-oriented paradigm, because this is the prevalent paradigm in the field of CSCW. As discussed in section 2.2.4 other techonologies such as knowledge-based systems can also belong into the field of CSCW; in this case the specification would have to include elements for *knowlegde representation* and *reasoning*.

All three aspects shall be discussed in the following sections, starting with the aspect labeled „inscription”. Consider the following example:

The CSCW-system ELISE (case study 2) includes a function that whenever a user views the list of articles in a journal, he is registered automatically as a reader of this particular issue. The information about who has already read a specific table of contents and who has ordered articles is displayed with the list of journals. So whenever a user logs into the system and the list of new journals is displayed, he can also see who has already viewed the table of contents. Figure 7 provides a screenshot of such a list.


Listen der individuellen Literaturarbeit
IV, neu
Artikel, noch nicht geliefert
IV, gelesen
Erledigte Bestellungen
Artikel, zu entscheiden
Listen der kooperativen Literaturarbeit
IV, mit Empfehlungen
IV, aus denen bestellt wurde
Empfohlene Artikel
Spezial-Listen
IV Suchabfrage definieren
A Suchabfrage definieren
GK
"gelesen"
"neu"

Nur neue / Alle: nur neue

Neue IV

ACM Transactions on Information Systems - TOIS (2005) Band 23, Heft 2

Gelesen von: RS MP Lus AC IJ CR TH | Empfehlungen an: | Bib - Bearbeitungsstatus: neu

Bst. durch: | IV eingestellt am: 22.06.2005 | Standort der Zeitschrift:

Empirical Software Engineering (2005) Band 10, Heft 3

Gelesen von: RS MP Lus AC IJ CR TH | Empfehlungen an: | Bib - Bearbeitungsstatus: neu

Bst. durch: Lus CR TH | IV eingestellt am: 22.06.2005 | Standort der Zeitschrift: Fernleihe

Zeitschrift fur Soziologie (2005) Band 34, Heft 2

Gelesen von: RS MP Lus AC IJ CR TH | Empfehlungen an: | Bib - Bearbeitungsstatus: neu

Bst. durch: IJ | IV eingestellt am: 22.06.2005 | Standort der Zeitschrift: Kopieren UB BO

ACM Transactions on Software Engineering and Methodology - TOSEM (2005) Band 14, Heft 2

Gelesen von: IJ MP | Empfehlungen an: | Bib - Bearbeitungsstatus: neu

Figure 7: Screenshot ELISE - List of Journals with Information about Readers

In this case the researchers first agreed to inform each other about their readings – this was an act of *ephemeral self-description*. Later when ELISE was being programmed, this act of self-description found its way into the requirements and became documented. And finally it was encoded in ELISE. The result is described in the example above.

The second aspect listed above was labeled „content”. The examples given in section 2.4.1 for self-descriptive documents include: organizational charts, organizational handbooks etc. Documentation of this kind is often stored in CSCW-systems, especially document-management-systems, for supporting cooperation. One could argue that the distinction between „inscription” and „content” is not clear cut; it might be a matter of degree. However from an organizational point of view it seems to be an important difference, whether self-descriptions are inscribed into the functionality of a CSCW-system, or whether they are created as „normal” documents or graphics outside a CSCW-system and then, after they are finished, merely stored in a CSCW-system.

The third aspect listed above is „socio-technical self-description”; it is the aspect on which this thesis focuses. Whenever an organization adopts a CSCW-system, it needs to readjust its self-description in a process of structural coupling to include the usage of the

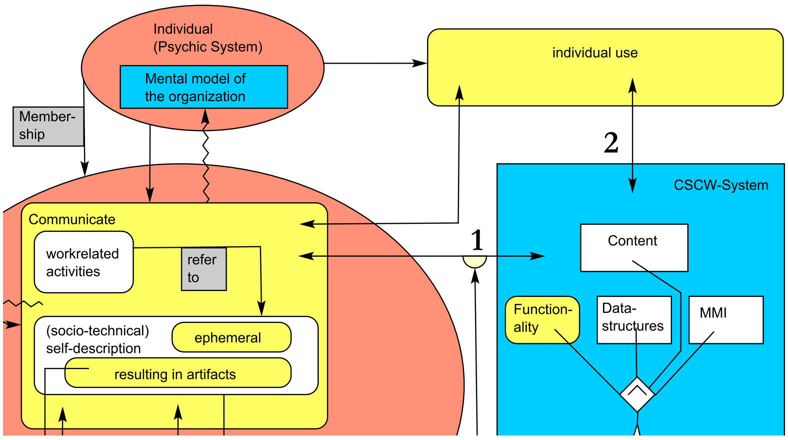
CSCW-system. Methodological support for this process includes creating and maintaining dedicated documents of socio-technical self-description.

The differentiation between these three aspects is not a merely analytical one. All three will occur within a CSCW-project and they are interdependent. A dedicated document of socio-technical self-description can be stored as a document within the same CSCW-system that it describes. The CSCW-system can provide functionality for displaying documents of socio-technical self-descriptions (cf. Example 32).

The need for distinguishing arises from the different needs for methodological support. Designing a CSCW-system that supports self-descriptions needs different methodological support than supporting an organization in adopting a CSCW-system by creating and maintaining documents of socio-technical self-description. The latter lies in the focus of this thesis.

3.1.4 Usage of CSCW-System

„Using" a CSCW-System



The circumstance that an organization „uses" a CSCW-system has been mentioned but not discussed so far. This section takes a closer look at the phenomena happening when a CSCW-system is „used" by an organization.

Although the verb „use" is used in an intuitive sense in many CSCW-related publications (e.g. [Robinson (1993a)], [Mark (2002)], [Prinz, Mark, Pankoke-Babatz (1998)]) it shall be clarified here how the term is defined for this thesis.

A CSCW-application is **used** if man-machine interactions are carried out as part of workrelated activities or if outputs of the CSCW-application are used for workrelated activities.

Term 16: Use of a CSCW-Application

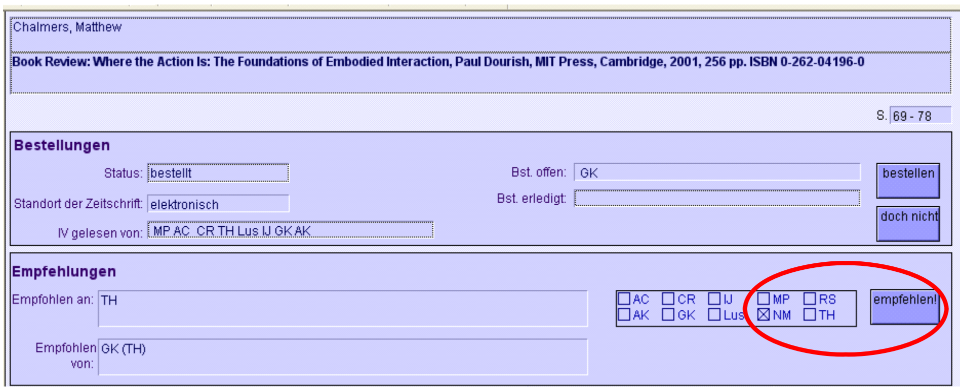
The definition includes the term man-machine interaction which refers to the following two definitions taken from [Herrmann (1999a), p. 43 (translation from German)]:

> *Interaction: Two systems interact if they influence each other. Phases in which a system behaves or changes states without being influenced by a particular second system alter with phases during which an influence is effective.*
>
> *Man-Machine-Interaction: A user's actions influence a machine by controlling it; and the machine also influences the user's behavior. During man-machine interaction the machine alters phases of automated control with phases of manual control. The machine's options for interaction and its limits are predefined.*

Communicate uses *CSCW-system*

The relation 1 denotes that *communicate* uses *CSCW-system*. This relation refers to the social usage of *CSCW-system* as opposed to the individual usage (2), which is discussed in the next section. The team selecting and ordering articles from scientific periodicals shall be used again to provide an example. The setting now is that the team uses a CSCW-system in which the tables of contents of all available journals are stored and that provides functionality for selecting, recommending and ordering articles (this setting is taken from case study 2 in which a CSCW-system named ELISE was developed and put into use; chapters 6 and 7 contain detailed descriptions and analyses):

One of the researchers selects one article from the open table of contents, marks the name of a colleague in the list and presses the button „recommend". The CSCW-system now attaches the recommendation to the article, and the colleague will see it in his list of recommended articles. Figure 8 contains a screenshot of the GUI.



Chalmers, Matthew

Book Review: Where the Action Is: The Foundations of Embodied Interaction, Paul Dourish, MIT Press, Cambridge, 2001, 256 pp. ISBN 0-262-04196-0

S. 69 - 78

Bestellungen

Status: bestellt

Standort der Zeitschrift: elektronisch

IV gelesen von: MP AC CR TH Lus IJ GK AK

Bst. offen: GK

Bst. erledigt:

Empfehlungen

Empfohlen an: TH

Empfohlen von: GK (TH)

☐ AC ☐ CR ☐ IJ ☐ MP ☐ RS
☐ AK ☐ GK ☐ Lus ☒ NM ☐ TH

Figure 8: Screenshot ELISE - Recommendation of Articles

In this example, the researcher uses the CSCW-system ELISE for the communicative act of recommending an article from the CSCW-journal to her colleague NM.

This communicative act shall be related to the definition of communication given in Term 9: the researcher uses ELISE for the activity *acts to convey information* by selecting the appropriate check box and clicking the right button. To complete the communication, the colleague NM then needs to use ELISE for her action *selects information from Alter's Action*.

This example is the basis for defining communicative use of a CSCW-application as follows:

The usage of a CSCW-system is defined as a **communicative use** if the usage occurs within the process of communication as defined in Term 9 and if both alter and ego use the CSCW-system.

Term 17: Communicative Use of a CSCW-System

And Figure 9 extends Figure 3 by including the CSCW-system.

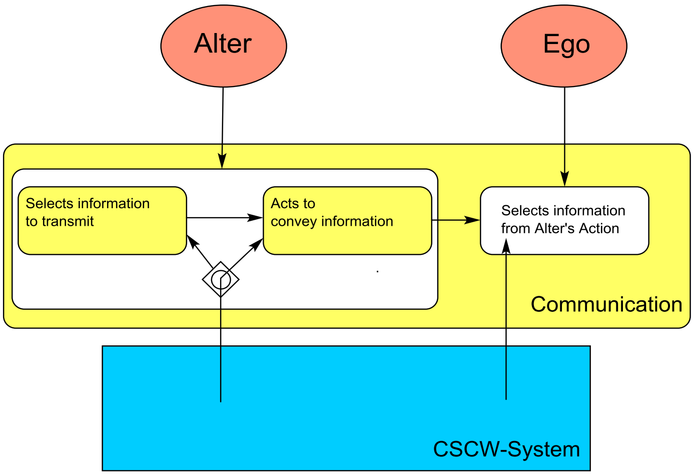

Alter
Ego
Selects information to transmit
Acts to convey information
Selects information from Alter's Action
Communication
CSCW-System

Figure 9: Communicative Use of a CSCW-System

Digression: Communication as Intentional Behavior

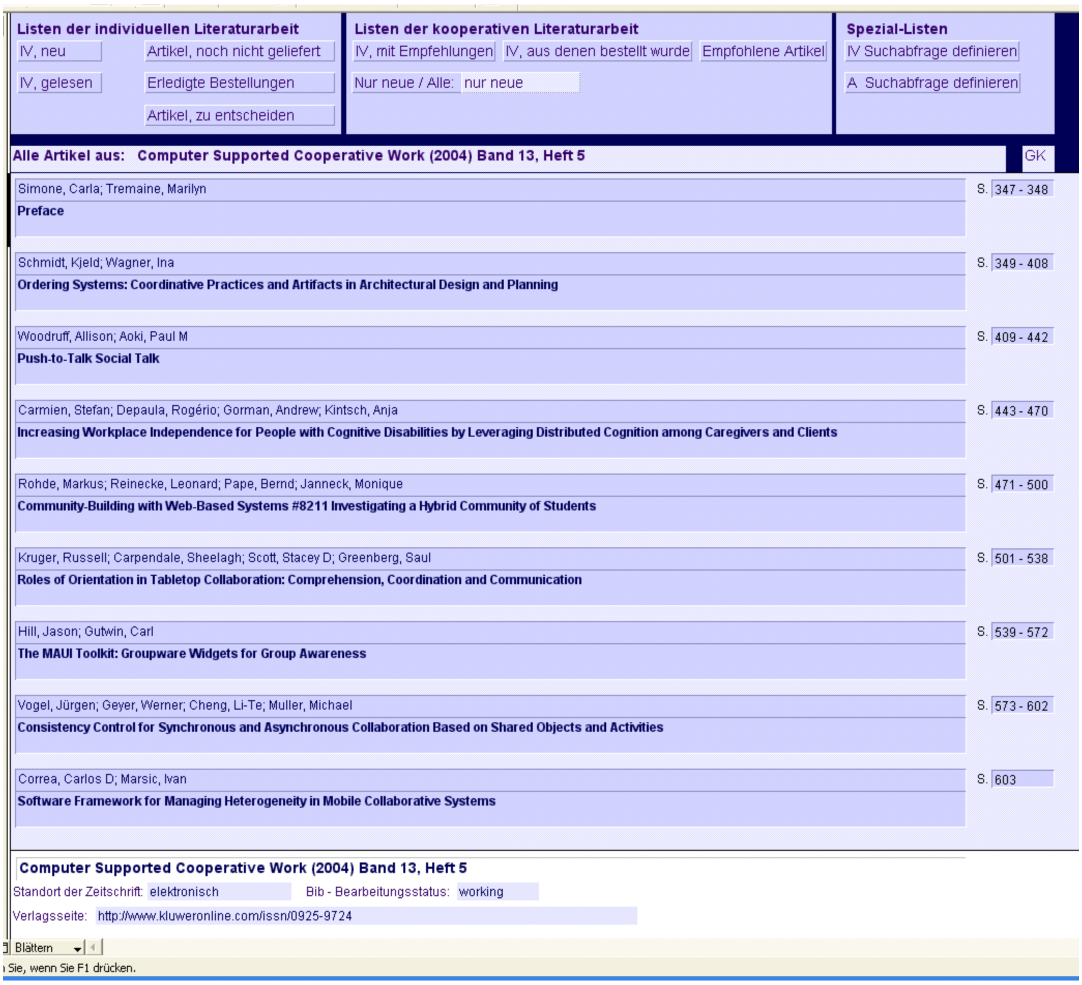
The discussion in the next section strives at distinguishing non-communicative (i.e. individual) use of a CSCW-system from the communicative use that is defined above. This distinction makes sense only if one assumes that not all behavior is communication. Therefore a brief digression into the theory of communication is taken at this point. Watzlawik is one representative of a theory that all human behavior has communicative aspects [Watzlawik et al. (1967)]. His famous insight that it is impossible not to communicate contains the essence of this approach [ibid. pp. 48].

Luhmann's conceptualization of communication is a quite different one. Not only does he request intentional behavior on the part of the sender (Alter's first two selections discussed in chapter 2.3.3 and illustrated in Figure 3); for a successful communication he requests that the addressee actually selects information from the actions he perceives. There are many theoretical approaches to the phenomenon of communication that lie in-between these two; an overview including a timeline is presented in [Kienle (2002), pp. 16].

In this thesis I follow those approaches that define action as intentional behavior and classify communication as action; i.e. communication is intentional. I do not regard non-intentional behavior as communication. The definition of communication given in Term 9 reflects these theoretical baselines. They are necessary for employing Luhmann's theories, but they are also in line with the context-oriented communication model presented in [Kienle (2002), p. 23] which has already proven its worth in designing CSCL-systems.

Individual *uses CSCW-system*

In contrast to the communicative use of CSCW-system, there is individual use which does not necessarily take place within the context of a communicative process. If someone uses a private field within a CSCW-system to annotate a document without sharing this annotation, then this usage is not communicative.


Listen der individuellen Literaturarbeit
IV, neu
Artikel, noch nicht geliefert
IV, gelesen
Erledigte Bestellungen
Artikel, zu entscheiden
Listen der kooperativen Literaturarbeit
IV, mit Empfehlungen
IV, aus denen bestellt wurde
Empfohlene Artikel
Nur neue / Alle:
nur neue
Spezial-Listen
IV Suchabfrage definieren
A Suchabfrage definieren
GK
Blättern
Sie, wenn Sie F1 drücken.

Alle Artikel aus: Computer Supported Cooperative Work (2004) Band 13, Heft 5

Simone, Carla; Tremaine, Marilyn **Preface**	S. 347 - 348
Schmidt, Kjeld; Wagner, Ina **Ordering Systems: Coordinative Practices and Artifacts in Architectural Design and Planning**	S. 349 - 408
Woodruff, Allison; Aoki, Paul M **Push-to-Talk Social Talk**	S. 409 - 442
Carmien, Stefan; Depaula, Rogério; Gorman, Andrew; Kintsch, Anja **Increasing Workplace Independence for People with Cognitive Disabilities by Leveraging Distributed Cognition among Caregivers and Clients**	S. 443 - 470
Rohde, Markus; Reinecke, Leonard; Pape, Bernd; Janneck, Monique **Community-Building with Web-Based Systems #8211 Investigating a Hybrid Community of Students**	S. 471 - 500
Kruger, Russell; Carpendale, Sheelagh; Scott, Stacey D; Greenberg, Saul **Roles of Orientation in Tabletop Collaboration: Comprehension, Coordination and Communication**	S. 501 - 538
Hill, Jason; Gutwin, Carl **The MAUI Toolkit: Groupware Widgets for Group Awareness**	S. 539 - 572
Vogel, Jürgen; Geyer, Werner; Cheng, Li-Te; Muller, Michael **Consistency Control for Synchronous and Asynchronous Collaboration Based on Shared Objects and Activities**	S. 573 - 602
Correa, Carlos D; Marsic, Ivan **Software Framework for Managing Heterogeneity in Mobile Collaborative Systems**	S. 603

Computer Supported Cooperative Work (2004) Band 13, Heft 5
Standort der Zeitschrift: elektronisch
Bib - Bearbeitungsstatus: working
Verlagsseite: http://www.kluweronline.com/issn/0925-9724

Figure 10: Screenshot ELISE - List of Articles within one ToC

There are other usages of CSCW-system for which it cannot be easily decided whether or not they are communicative. Consider the researcher who uses the CSCW-system

ELISE to skim through a list of articles without ordering any article (Figure 10 shows a screenshot of the GUI). On the one hand one could argue that this is no communicative action because he is alone and there is no connecting activity by anybody else. On the other hand one could argue that this researcher is the recipient of a communicative act that was started by the student worker who imported the articles into the database. In terms of Luhmann's concept of communication (cf. chapter 2.3.3, Figure 3) the student worker is *Alter* who informs about new articles (*selects information to transmit*) by importing them into the CSCW-system ELISE (*acts to convey information*). The researcher is *Ego* who receives the information (*selects information from Alter's action*). In this interpretation the researcher's usage of the CSCW-system would be communicative.

There are more examples which cannot clearly be categorized into communicative or not: If the researcher wanted to recommend one of the articles in the list shown in Figure 10 to a colleague, he would have to click on the title of the article, which would lead him to the screen shown in Figure 8 there he would have to select the name and press the button „recommend". On the one hand one could argue that all of these steps belong to step two of a communication (*acts to convey information)*, on the other hand one could say that only the last one (actually pressing the button „recommend") is the communicative one. If the researcher changed his mind and stopped the sequence of interactions before actually recommending the article, then no communication would have taken place. So for some interaction between user and CSCW-system one can only decide retrospectively whether or not it was part of a communicative act.

The lesson to be learned from these examples is that not all usage of CSCW-system is social; at least not intentionally from the very beginning. Therefore it does make sense to distinguish between individual and social use of CSCW-system in a model aiming to explain the relations between organization and CSCW-system.

While it might be possible to classify interactions retrospectively on the basis of a concrete model of communication, such a classification cannot take place in parallel with the users' actions. This is true especially because some interactions may change their characteristic as a result of other actions.

The action *individual use* is included in the model for two reasons: First, individual use is a relevant issue with regard to designing supportive functions and with regard to training. Second, *individual use* may lead to *Content* in the CSCW-system, that was not meant to be shared in the beginning, but that is later used by other communicative acts. Therefore the *individual use* is closely intertwined with the social use and needs to be regarded. Summarizing, the following shall be defined.

> The usage of a CSCW-system is defined as **individual use** if the actor does not regard his actions as part of a communicative process as definded in Term 9.

Term 18: Individual Use of a CSCW-system

3.2 Deriving Socio-technical Concepts

3.2.1 Basic Definitions

The interdependencies of software-systems and organizations lead technical disciplines such as software engineering or information system design to the definition and usage of the term „socio-technical" for their purposes (cf. section 2.1.3). Shortcomings of existing conceptualizations were identified (cf. section 2.1.5) and concepts of newer systems theory were introduced to provide a base for a reconceptualization; Table 2 summarizes the results. Figure 6 illustrates the complex setting that emerges when an organization uses a CSCW-system. The discussions of the subsections above are now used for deriving definitions for socio-technical system, socio-technical organization, socio-technical setting, and socio-technical self-description.

Following the notion that a socio-technical system consists of two "*independent but correlative"* [Cummings, Srivastva (1977), p. 60] parts and guided by Luhmann's statement that systems are composed of homogeneous parts (cf. section 2.3.7, p. 45), my definition of a socio-technical system does not integrate technical and social systems into one larger type of system; rather they belong to the environment of each other and the concept of structural coupling (cf. Term 10) is used to describe how they form special relations to each other. A logical consequence of the theoretical discussion would be the following definition of a socio-technical system:

A social system that maintains a structural coupling with a technical system is called a **socio-technical system** with respect to this technical system.

Term 19: Socio-Technical System

With this definition, a socio-technical system would be a special type of social system and would not include the technical system. Rather, the technical system remains in the environment of the social system and the concept of structural coupling is used to describe the special relationship.

The advantages of this definition are:

- It is in line with the theoretical base of system theory that self-referential systems cannot comprise different types of systems (cf. section 2.3.7, p. 45).
- It provides for the description and methodological support of the special relationship between a social and a technical system.
- By detailing and quantifying the structural coupling it is possible to distinguish between different types and degrees of socio-technical systems.
- It allows describing socio-technical systems as subsystems of social systems. In this way a department within a company can be described as a socio-technical system while the company as a whole is not.

The major disadvantage of this definition is that it is against the established common understanding that a socio-technical system is a heterogeneous system comprising both social and technical system. In order to avoid confusion but to keep the advantages of the reconceptualization at the same time, I use the term socio-technical organization instead.

An organization that maintains a structural coupling (cf. Term 10) with a technical system is called a **socio-technical organization** with respect to this technical system. The following characteristics are true for socio-technical organizations in the special case of CSCW-systems:

1. Within a socio-technical organization there is communicative use of the CSCW-system in the sense of Term 17.
2. A socio-technical organization maintains socio-technical self-descriptions (cf. Term 21).

Term 20: Socio-Technical Organization

Since organizations are a special subtype of social systems (cf. section 2.3.6) socio-technical organizations can be regarded as a special subtype of socio-technical systems. It is quite adequate to concentrate the focus of this thesis on organizations because CSCW typically takes place in organizations. Socio-technical self-descriptions which are included in the definition of socio-technical organizations are defined as follows (in extension of the preliminary definition in Term 13):

Socio-technical Self-descriptions are communications by which a socio-technical organization maintains its boundaries. The following characteristics apply to socio-technical self-descriptions in the special case of CSCW-systems:

1. Socio-technical self-descriptions describe which communications (actions) are acceptable within an organization and how they relate to the CSCW-system.
2. Socio-technical self-descriptions are used by other communications (actions) as guidance.
3. Socio-technical self-descriptions take the form of ephemeral communications, durable artefacts such as documents and inscriptions in the CSCW-system.
4. Socio-technical self-descriptions can and should be distinguished from "normal" communications (actions), because this increases the organization's chances for developing more demanding and specialized structures with respect to the CSCW-system.

Term 21: Socio-Technical Self-Description

Although the term “socio-technical system” is not used for subsuming organizations and CSCW-systems into one larger type of system, it is indeed helpful to have a term which refers to the whole complex. I will use the term socio-technical setting for this purpose.

The complex created by a socio-technical organization and the relevant CSCW-system shall be called **socio-technical setting.** Figure 6 illustrates the components and their relationships.

Term 22: Socio-Technical Setting

My term of socio-technical setting based on Luhmann’s system theory is the counterpart to the concept of socio-technical system as it was coined by the early Tavistock researchers based on system theory of their time.

Structural Coupling, Adoption and Appropriation

The term „structural coupling” (cf. Term 10) taken from Luhmann’s system theory (who himself has adopted it from Maturana and Varela) is used to explain the characteristics of the changes taking place within an organization with respect to a CSCW-system. The concepts adoption and appropriation that are commonly used within the field of CSCW are closely related and shall be discussed at this point.

In [Hoffmann (2004), p. 37] several concepts of acceptance and adoption of technical systems are discussed and the following definition of adoption is derived [ibid., translated from German]:

> *„Adoption is a process performed by users, which results in a lasting inclusion of the technology into individual or social patterns of behavior.”*

Hoffmann then describes seven subprocesses included in the overall process of adoption. One of these subprocesses includes the adaption of behavior, especially the development of effective patterns for cooperating and using the technology.

The term appropriation is used by DeSanctis and Poole as one component of their adaptive structuration theory. With reference to Ollman, they define the term „appropriation of technology” as *„the immediate, visible actions that evidence deeper structuration processes”* [DeSanctis, Poole (1994), p. 128]. The *„structuration processes”* refer to processes of change within an organization, which take place in response to a new groupware.

Dourish deliberately chooses a more practical definition and states [(2003), p. 467]:

> *„Appropriation is the way in which technologies are adopted, adapted and incorporated into working practice. This might involve customization in the traditional sense (that is, the explicit reconfiguration of the technology in order to suit local needs), but it might also simply involve making use of the technology for purposes beyond those for which it was originally designed, or to serve new ends.”*

The concepts of adoption, appropriation and structural coupling share one important tenet: the organization's autonomy in using a CSCW-system. All three concepts assume that processes within an organization rather than external design activities are responsible for the actual incorporation of a CSCW-system into work procedures. The question is whether the term "structural coupling" could be replaced by either "appropriation" or "adoption", which are already known within the field of CSCW. For this thesis, I prefer using the term "structural coupling".

First, because it is an integral part of the theoretical concepts I am grounding my work on. Second, because the field of CSCW provides commonly agreed definitions neither for appropriation nor for adoption. And third, because the definitions cited above suggest that structural coupling is the broader concept with regard to processes of organizational change. While the concepts of appropriation and adoption describe whether and how an organization makes use of a certain technical system, the concept of structural coupling includes a wider range of organizational changes. Consider the following example taken from case study 1 "mobile communication system": One purpose of the new mobile communication system SpiW-Com was to reduce the number of phone calls by truck drivers that interrupt the dispatchers' work. Therefore drivers and dispatchers discussed rules when they would use the phone and when they would use SpiW-Com (cf. Example 14, p. 186). In the end they decided to reduce the usage of mobile phones significantly.

The organization's changed behaviour with regard to the usage of mobile phones is clearly part of its structural coupling with the new CSCW-system SpiW-Com. The debate was important, because without it, the drivers would have used SpiW-Com in addition to the usual phone calls. For supporting the evolvement of socio-technical organizations the notion of organizational change needs to exceed the direct adoption or appropriation of a CSCW-system. Structural coupling might be a good concept for describing this broader sense of organizational change.

Socio-technical Organization or not? – Examples

To summarize the abstract discussions and definitions of the previous sections, some examples shall be provided for elucidating the concept of socio-technical organizations.

1) A member of a research team uses the group's document management system for storing the bibliography of her thesis. This is not an example of a socio-technical organization, because there is no communicative use (cf. Term 17) of the document management system.

2) A research team uses a CSCW-application for informing each other about their readings and recommending articles to each other (cp. case study 2). This is an example of a socio-technical organization in the sense of Term 20.

3) Vendors and customers of E-Bay do not form a socio-technical organization because the vendors and customers are no organization.

At this point, the theoretical considerations yielded all the constructs necessary for describing phenomena occurring in the context of CSCW-projects. The next step is to derive methodological support for such projects.

3.2.2 Implications for CSCW-Projects

Definition of CSCW-Projects from a Socio-Technical Perspective

Table 2 already summarizes the implications derived from the theory of social systems for methods supporting CSCW-projects. With the conceptualizations of the previous sections, the goals and characteristics of CSCW-projects can be further detailed.

CSCW-projects lead to socio-technical settings (cf. Term 22); for this two interrelated subtasks have to be handled:

a) The technical CSCW-system (cf. Term 7) has to be designed and implemented.

b) The evolvement of a socio-technical organization (cf. Term 20) has to be supported.

Term 23: CSCW-Project

Process Model for CSCW-Projects maintaining a Socio-Technical Perspective

The insights summarized in Table 2 shall also be described in a process model (Figure 11) that also illustrates the focus of this thesis.

The process model includes a process *CSCW-Project* with the two subtasks named in Term 23, *Support organizational change / structural coupling* and *System design*. The half circle at the bottom of the rounded rectangle indicates that there are more subtasks to a *CSCW-Project*; depending on the way the project is organized, these can include project management tasks, early feasibility studies etc. My focus in this thesis is the integration of organizational change and software engineering and the resulting methodological components should be usable within different forms of projects. Therefore as few assumptions as possible are made concerning the overall project organization; the limitation to two relevant subprocesses within *CSCW-Project* reflects this goal. The *CSCW-Project* as a whole is responsible for creating a *Socio-technical Setting* (cf. Term 22). The *Socio-technical Setting* in the process model is a reduced form the socio-technical setting illustrated in detail in Figure 6 (the half-circle at the bottom of the rectangle symbolizes the omissions). It contains the *Socio-technical Organization* (cf. Term 20) including *work procedures*, which make use of a *CSCW-Application* (cf. Term 8) that is embedded in a larger *CSCW-System* (cf. Term 7).

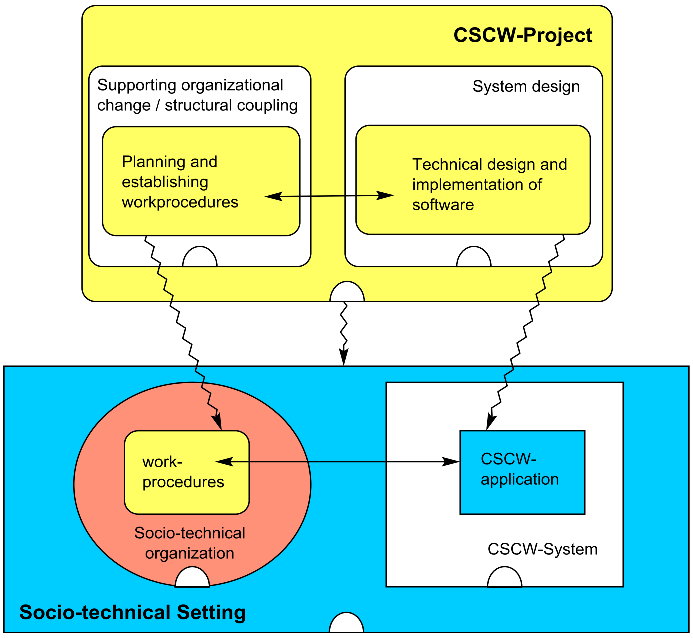

CSCW-Project
Supporting organizational change / structural coupling
System design
Planning and establishing workprocedures
Technical design and implementation of software
work-procedures
Socio-technical organization
CSCW-application
CSCW-System
Socio-technical Setting

Figure 11: Process-Model for CSCW-Projects

With regard to my focus, two more detailed tasks are explicitly illustrated and thereby emphasized: *Planning and establishing work procedures* as the one part of organizational change on which this thesis focuses. Obviously organizational change comprises more than just work procedures; the organizational culture and the hierarchical structure are other aspects that could be subject to processes of organizational change. But since CSCW-systems directly influence the way people go about their daily work, and since the interdependencies between technical design and organizational change are most obvious in the work procedures, I choose this topic. *Technical design and implementation of software* is the second task that is emphasized. Embedded in a larger system design [Sommerville (2004)] there is software design and with respect to technical design and implementation, I concentrate on software.

While the overall *CSCW-Project* is responsible for creating the overall *Socio-technical Setting*, it is important to note that the *Work procedures* are created by the process *Planning and Establishing Work procedures* while the *CSCW-Application* is created by the process *Technical design and implementation of software*. This separation reflects the insights elaborated and discussed in chapter 2.1.5 (p. 31) that autopoietic organizations and allopoietic software need different methods for change and design respectively. This separation does not imply that there may not be activities within the project that are relevant for both processes.

By explicating both parts of the bipartite character of a socio-technical setting, the project team of a CSCW-project should be supported in keeping a socio-technical perspective throughout the whole project.

Project vs. Design in Use

It is a widely accepted notion within the field of CSCW, that the design of a CSCW-system does not end once the software is implemented [Henderson, Kyng (1991)].

Concepts like appropriation (cf. section 3.2.1, pp. 83), metamorphosis [Orlikowski (1996)] or evolving use [Andriessen, Hettinga, Wulf (2003)] describe how the users shape the software by inventing new and sometimes unanticipated ways of using it. Design concepts like end-user development [Wulf, Jarke (2004)] or meta-design [Fischer et al. (2004)] seek to enable the user to continue software design after it has gone operational. Does my focus on projects rather than processes neglect the insight that design and use are intrinsically interwoven? No. I choose the setting of a project, because most undertakings to develop and deploy new software take the form of projects; and my goal is to provide further methodological support for such undertakings. My argument is that the interplay of (technical) design and use can start prior to the actual usage of a CSCW-system.

Parallelizing Structural Coupling and Software Engineering

The process model in Figure 11 suggests parallelizing the processes of structural coupling and technical software engineering. This suggestion is not self-evident, therefore some argumentation is provided in this section.

The integration of processes of organizational change and software engineering adds complexity to the projects. One could argue that, after a thorough phase of requirements elicitation and an initial design, technical software engineering could take place and the integration into the users' work procedures does not start until the implementation is completed. Process models such as the V-model do indeed suggest such an approach (cf. section 4.1.3). My approach is that the process of structural coupling within the organization can start by preparing new socio-technical self-descriptions before the software is completed. Three arguments for this proposition shall be discussed:

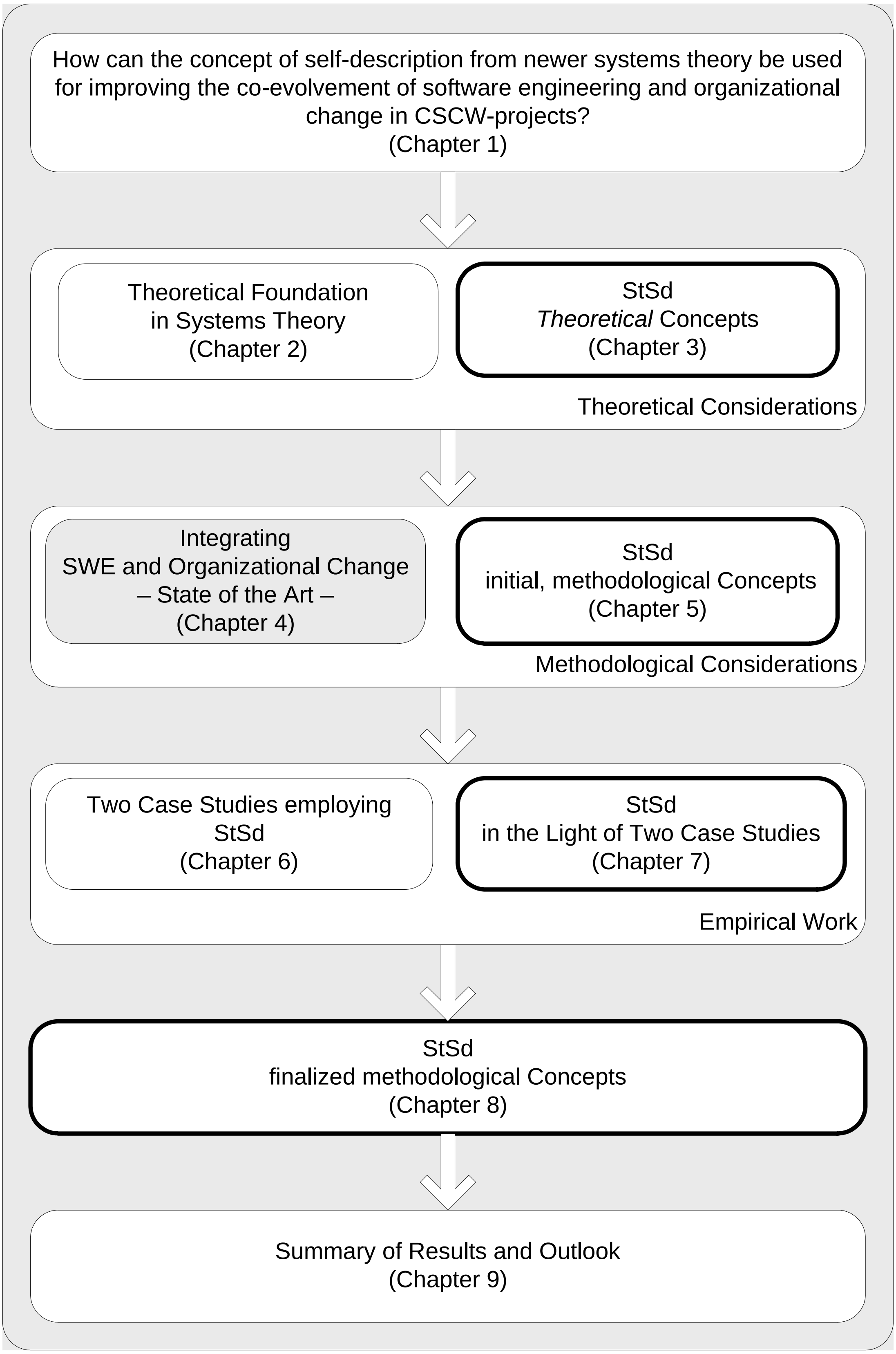
First, there is a de-facto parallelism. Even after a thorough phase of requirements elicitation there will be decisions to be made during detailed design and implementation that bear implications for the work procedures. If there are actors involved in a parallel process of structural coupling, then there is a process within the CSCW-project that can adequately respond to organizational questions. Otherwise the decisions would be made anyway but without involvement of the organization.

Second, evolutionary methods of software engineering are common for software such as CSCW-applications (cf. section 4.1.3). They foresee the incremental implementation and delivery of software. The parallelism of structural coupling and technical software-development is a necessity in this case. How should a software-version be adequately evaluated if the work procedures into which it is embedded are not documented and discussed within the organization?

Third, it is expected that initiating the process of structural coupling prior to use saves time and problems in the actual deployment phase. Basic conventions can already be discussed and agreed to, before the software is actually used.

The suggestion of parallelizing structural coupling and software-development should not be taken as a statement neglecting the process of appropriation that takes place in the course of using a CSCW-application (cf. section 3.2.1, p. 83). The suggestion is that such processes can start earlier in a project if they are methodologically well supported.

The discussion of the case studies will show that it was indeed possible to plan work procedures and conventions ahead of time.



How can the concept of self-description from newer systems theory be used for improving the co-evolvement of software engineering and organizational change in CSCW-projects?
(Chapter 1)

Theoretical Foundation
in Systems Theory
(Chapter 2)

StSd
Theoretical Concepts
(Chapter 3)

Theoretical Considerations

Integrating
SWE and Organizational Change
– State of the Art –
(Chapter 4)

StSd
initial, methodological Concepts
(Chapter 5)

Methodological Considerations

Two Case Studies employing
StSd
(Chapter 6)

StSd
in the Light of Two Case Studies
(Chapter 7)

Empirical Work

StSd
finalized methodological Concepts
(Chapter 8)

Summary of Results and Outlook
(Chapter 9)

4 Integrating SWE and Organizational Change – State of the Art

Chapter 3 elaborated my theoretical concepts of socio-technical settings, socio-technical organizations and socio-technical self-descriptions; it ended with implications for the methodological support of CSCW-projects. The next step is to check existing methods for their potential support of CSCW-projects as they are defined in Term 23. For this purpose, chapter 4 analyses different approaches for integrating software engineering and organizational change. Process models and methods from software engineering (section 4.1), systemic intervention (section 4.2) and design methods developed in the field of CSCW (section 4.3) are discussed in the light my theoretical concepts. The goal of this analysis is to identify in how far existing methods support the structural coupling between an organization and a CSCW-system, and whether support for the creation and maintenance of socio-technical self-descriptions can be found. A summative discussion (section 4.4) identifies best practices as well as methodological gaps.

4.1 Software Engineering

4.1.1 Introduction

Although software engineering is often understood as an engineering discipline (cf. section 2.2.3), there are many voices emphasizing the close relationship between social processes and software engineering [Floyd, Züllighoven (2002), p. 765]. This point of view is especially prominent for the early phases of a software engineering project, when the new software has to be envisioned and designed by both future users and the software engineers. An overview of „socio-technical" methods for requirements elicitation can be found in [Atkinson (2000)], and some of the methods mentioned there will be discussed in more detail in section 4.3.3.

But what is the relation between *Technical design and implementation of software* and the process of *Planning and establishing work procedures* in Figure 11? The definition of software engineering given in Term 6 indicates that the task of supporting organizational change is not included. If however, as suggested in Figure 11, the two processes are interrelated within a CSCW-project, then there should be interfaces and links between the two. The following sections discuss the relationship between software engineering and organizational change with respect to methodological support. This discussion results in two lists (cf. Table 11, section 4.4): The first contains software engineering methods that could provide a link to *Supporting organizational change / structural coupling*; the second contains assumptions that underlie prominent approaches to software engineering but that are problematic with respect to organizational change.

4.1.2 Tasks within Software Engineering

Four Generic Tasks

In [Sommerville (2004), p. 64] software engineering is described to contain the following tasks:

1. *Software specification: The functionality of the software and constraints on its operation must be defined.*
2. *Software design and implementation: The software to meet the specification must be produced.*
3. *Software validation: The software must be validated to ensure that it does what the customer wants.*
4. *Software evolution: The software must evolve to meet changing customer needs.*

According to Sommerville these four tasks are needed for software engineering independent from the process model that is applied.

Deployment as a Task of SWE

Sometimes the process of deployment (i.e. the tasks necessary to install the software at the customer's site and bring it into operation) is also considered as a separate task of software engineering: the rational unified process (RUP) includes a workflow for deployment ([Kruchten (2004), p. 237] and the V-modelXT also includes deployment [Broy, Rausch (2005)]. For CSCW-systems in organizations, a professional deployment of the software is essential: often the number of client-systems on which software needs to be installed is so large that special processes need to be organized; when a server-system is updated on which many clients rely, this has to be carefully planned and synchronized with organizational needs. Due to its relevance for CSCW-projects, I consider deployment as a task of software engineering; the way deployment is methodologically supported by the RUP is discussed in section 4.1.7.

4.1.3 Process Models for Software Engineering

Process models organize how the tasks necessary for software engineering are organized within a project. According to [Summerville (2004), p. 64] all process-models for software engineering contain the tasks listed above; they differ in the chronological order of the tasks and in the way the tasks are linked to each other by intermediate artifacts such as documents.

Linear Approaches

Linear approaches such as waterfall-models foresee no overlapping of phases; for each phase they define a set of deliverables that needs to be completed and accepted before the next phase may commence [Sommerville (2004), p. 9]. In Germany an important representative of linear approaches is the so-called V-model which is a de-facto standard for public software-contracts ([Klink, Oesterreich (2005), p. 55], [Broy, Rausch (2005)]). There are two characteristics of linear process models that are critical with respect to CSCW-projects.

First, the assumption that requirements for the CSCW-system can be definitely established after an initial requirements elicitation phase is not adequate for CSCW-projects. The process of structural coupling will inevitably bring about new or changed requirements, thus a process model that handles evolving requirements as normal rather than exceptional is better suited for CSCW-projects. The new version of the V-model, now named V-model XT, has reacted to this need by offering two different project strategies: an incremental one in which the requirements are completely described in the beginning of the project and an agile one that allows a stepwise refinement of specification as well as development and delivery [Klink, Oesterreich (2005), p. 57].

Second, linear approaches such as the V-model foresee user participation in the very beginning and in the end only. In the beginning users are involved in the process of requirements elicitation and in the end they are participants of acceptance testing and training courses. The phases in-between (technical design, implementation, testing) belong to the software engineers and foresee no further user participation. Such a strategy does not harmonize with a process model for CSCW-projects that foresees the parallelizing organizational change and software development (cf. Figure 11).

Since continuous user involvement was identified as one of the key success factors for software engineering (Standish Group, cited in [Moll et al. (2004), p. 421]), there are modifications of linear process models that include an additional and parallel process "user involvement" [Moll et al. (2004), p. 427]. This idea matches with the process model for CSCW-projects and I will take it up: the users are continuously involved in reviewing and evaluating the CSCW-system that is being developed. However, the purposes of user involvement addressed by employing socio-technical self-descriptions transcend those intended by Moll et al. Moll et al. described the phase and purpose of user involvement as follows ([ibid. p. 428], translated from German, emphasis added):

> *The phase "user involvement" runs in parallel with the phases design and implementation. It creates* **all prerequisites** *which are necessary for a successful deployment of the software system with the users: user manuals, training etc.*

While user manuals and training are important activities to prepare the individual use (cf. Term 18) of a CSCW-system, they do not suffice for preparing the communicative use (cf. Term 19) and its integration into work procedures. For the latter, I suggest supporting the process of structural coupling (cf. Term 10) in parallel with the technical software development. For this purpose, evolutionary approaches seem to be more appropriate than linear ones.

Evolutionary Approaches of SWE

Evolutionary software development allows the software specification to evolve throughout a project. It can be defined as follows [Sommerville (2004), p. 65]:

> *„This approach interleaves the activities of specification, development and validation. An initial system is rapidly developed from abstract specifications. This is then refined with customer input to produce a system that satisfies the customer's needs."*

Evolutionary approaches to software engineering, as they are described in this definition, are better suited for integrating organizational change and software engineering than linear process models are. If the software is developed using a process of stepwise refinement, then this process can continuously provide links to the process of *Planning and establishing work procedures* (cf. Figure 11). Two aspects that enable this integration of technical and organizational development shall be discussed at this point.

First, intermediate artifacts: the process of stepwise refinement produces a series of intermediate artifacts such as prototypes. These are the basis for the collection of feedback from the users. Such an approach links well to the idea that the structural coupling within an organization is an ongoing, communicative process. As the software-development project proceeds, new artifacts can be given to the communicative processes of the organization and thereby serve as a link between technical and organizational development. The concept of socio-technical self-description makes use of such intermediate software-artifacts by integrating them with descriptions of work-procedures (cf. 7.2.2) and organizing workshops in which the combination of both is subject to discussion (cf. 7.3). These workshops support processes of organizational change, because they do not only view the work procedures to be a source of input for requirement engineering but as something that needs to evolve in parallel with the software engineering.

Second, ability to react to changes: requirements for a CSCW-system are never really frozen instead they change and evolve in the course of a project. One advantage of evolutionary approaches is that they do not handle such changes as exceptions [Kruchten (2004), pp.55]. This point of view links well to the process of structural coupling. Within an organization the process of structural coupling to the new software system is a communicative process of change during which the members of the organization learn about the software. It is the normal course of events that new requirements come up or existing requirements change. Any software engineering method that has provisions for this is more suitable to link to processes of organizational change than those that are not.

One example of an evolutionary process model for software engineering is the STEPS model discussed in section 4.3.3 (p. 121).

Agile Methods and Extreme Programming

The very formal, often linear, process models for software engineering that foresee a large amount of documentation to be written (e.g. [Parnas, Madey (1995)]) were a response to the „software-crisis" manifesting itself in projects that took much longer than planned, cost must more than budgeted and did not necessarily deliver the software needed by the customers. With the so-called „agile methods" of software engineering (e.g. [Webpage Agile Manifesto], [Sommerville (2004), pp. 396], [Cockburn (2002)]) the pendulum swings back towards less rigid and more adaptable process models, a reduction of the overhead documentation and a concentration on the software to be developed. Three aspects which are relevant with regard to CSCW-projects shall be discussed.

First, it should be noted that agile methods foresee an incremental delivery of the software to the customer [Sommerville (2004), p. 397]. Therefore agile methods share the advantages of evolutionary approaches with respect to their integration with processes of organizational change.

Second, extreme programming [Beck (2004)] – probably the most prominent agile method – foresees that a representative of the customer's organization is on site during the development of the software (e.g. [Cockburn (2002)], [McKenzie, Monk (2004)]). His task is to *„provide and prioritize new system requirements and to evaluate the iterations of the system."* [Sommerville (2004), p. 397]. Such a method is an example of how the user's knowledge is continuously made available for software engineering; and it is a good anchor point for the integration of technical development and organizational change processes. However, the cooperation of one user with the software-developers does not automatically trigger communications within the customer's organization, which are needed for organizational change (cf. section 2.3). It was discussed in section 2.3.7 that the structural coupling of an organization with a CSCW-system is a social process within the organization. For supporting organizational change, the aspect of user-involvement would need to be extended, e.g. by transferring information from the software-development into the organization and triggering communication about the CSCW-system and its usage there. For this purpose, the methodological concept of socio-technical self-descriptions combines intermediate artifacts from software engineering (e.g. prototypes) with diagrams representing work procedures and makes them available for discussion within the organization. Here first options of relating agile methods for software engineering with methods of supporting organizational change become visible. But this leads also to a potential conflict of methods, which is the last aspect to be discussed for agile methods of software engineering.

Third, agile methods strive to reduce the amount of overhead documentation. The concept of socio-technical self-description however, relies on self-descriptive artifacts that support the organization in changing its workprocedurs. This can be interpreted as a contradiction, because an additional type of documentation is suggested. However the essence of agility is its adaptability to local needs [Cockburn (2002)]. If the creation and maintenance of self-descriptive artifacts for fostering the discussion, change and establishment of technically supported work procedures, is accepted as a suitable means within CSCW-projects, then the methods I suggest are in line with agile concepts, and both could be combined as suggested in section 8.3.

RUP and UML

The Rational Unified Process (RUP) is one process model for software engineering projects (e.g. [Sommerville (2004), pp. 82], [Kruchten (2004)]). The RUP uses the categories of phases, disciplines / workflows, and iterations to organize software engineering projects. It foresees nine disciplines that are iteratively carried out over four project phases. Figure 12 is a standard description taken from [Kruchten (2003)].

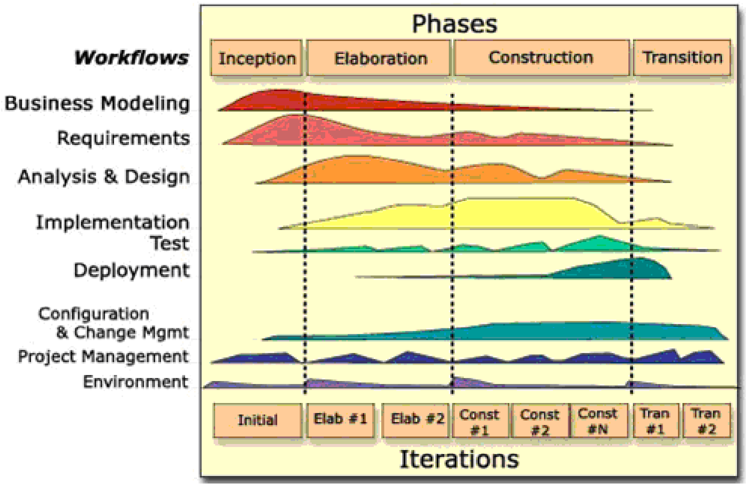


Figure 12: The Rational Unified Process (RUP)

[Kruchten (2003)]

The rational unified process is a software engineering process that extensively employs visual models for the specification of software. For this the Unified Modeling Language (UML) is used which provides a set of models [Rumbaugh, Jacobson, Booch (2005), pp. 26] that can be used to represent a software-system from different perspectives and at various levels of detail. However, the UML explicitly provides no procedures for the development of software [Rumbaugh, Jacobson, Booch (2005), p. 10]. These are added by process models such as the RUP which understands itself as *„a guide to the effective use of the UML for modeling"* [Kruchten (2004), p. 28].

Although the RUP is a commercial product now, it has its roots in the history of software engineering methods such as the iterative approach to project management going back to Boehm‘s spiral model [Boehm (1988)] and the usage of use-cases as suggested by [Jacobson (1993)].

In the following sections more methodological aspects of software engineering and their relation to processes of organizational change are discussed. Because the UML is a „de facto standard" in software engineering today [Sommerville (2004), p. 155] and because the RUP is a generic process model for the UML, the discussions focus on RUP and UML but also include other methodological approaches where needed.

4.1.4 Visual Models in SWE

Visual models such as Petri-net-diagrams, data-flow models, or entity relationship diagrams, to name a few, are common tools in software engineering. Again, the discussion here concentrates on those aspects of the usage of diagrams that bear potential relations to issues of organizational change.

Affordances and Limitations of Diagrams

Most of the documents produced within the framework of the RUP are graphical models that represent the software to be produced from different perspectives. With regard to the construction of software as a technical system (cf. section 2.2.2) these diagrams are used as construction plans. They contain a complete and unambiguous description of the software [Kruchten (2004), p. 82] that eventually leads to the software-code. But they are also used by activities summarized as *socio-technical* activities in Figure 11: Within the RUP, the diagrams are used for supporting the discussion among different groups involved in the process of software engineering. Undoubtedly, diagrams are artifacts well suited for supporting communication processes in groups; however there are two assumptions about *„unambiguous communication"* [Kruchten (2004), p. 12] and unambiguous specification of behavior [Kruchten (2004), p. 13] that are problematic in the light of newer systems theory.

Kruchten claims that *„Using a standard modeling language such as the Unified Modeling Language (UML), members of the development team can unambiguously communicate their decisions to one another."* [Kruchten (2004), p. 12]. Software-developers working on a project form a social system and are subject to all the principles applicable to such systems. Especially their communication is never unambiguous; rather it is characterized by the phenomenon of double contingency (cp. Section 2.3.3). The implications that can be drawn from this insight about the organization of design teams and design processes themselves are not my focus. But since I will make extensive use of diagrams for the communication between software-developers and users as well as in-between users, it should be noted at this point that even though diagrams can be useful artifacts for supporting communicative processes, they are not able to override the basic principles of social systems: communication remains highly ambiguous even with the usage of diagrams.

Another claim Kruchten makes is that *„Use cases and scenarios unambiguously specify behavior."* [Kruchten (2004), p. 13]. Similar remarks as made with respect to the ambiguity of communication apply here. Use-cases contain descriptions of the actions of human actors as well as of processes performed by the software. For the latter one could think of unambiguous descriptions although the informal nature of use-case descriptions [Kruchten (2004, p. 100]) suggests that they might not be completely unambiguous. For the human actor it is simply not possible to unambiguously prescribe his or her behavior. This insight does not render visual models useless, but when considering using them as a means to link software-development to processes of organizational change one must keep it in mind.

This will not be the last time that the limitations – or even the risks – of diagrams representing aspects of cooperative work are discussed. Within the field of CSCW there is a great skepticism against the usage of formal diagrams representing work procedures; for a further discussion, refer to section 4.3.5. The methodological support I develop embeds the usage of graphical models into moderated communicative processes to cope with the problems related to representations of work procedures (cf. 5 for a presentation of the concept and section 7.3 for a discussion of the experience made during the case studies).

Extending the Concept of Multiple Perspectives by Self-descriptive Artifacts

Although the visual models and the concept of multiple perspectives provide good links between the RUP and the UML on the one side and processes of organizational change on the other, some gaps will have to be closed. The purpose of the UML-diagrams is to describe the technical system; organizational aspects are added as long as they are needed for describing the behavior of the software (e.g. sequence diagrams including users' actions as well as software processes). Even the use-cases have the purpose of defining the boundaries of the technical system and not of describing options within the organization (see next section for details).

One way of extending the UML-concept of multiple perspectives towards supporting organizational change, would be adding artifacts of socio-technical self-description as another type of diagram. But such diagrams underlie the principles discussed above and would require new or changed workflows in the RUP. The case studies presented in chapters 6 and 7 show how artifacts of socio-technical self-descriptions can be included in a process of software engineering. A discussion of the potential combination of methods is included in section 8.3.

These thoughts concerned visual models in general. Two kinds of UML-diagrams bear especially obvious links to processes of organizational change: Use-case diagrams and business models. The following two sections explore the possibilities of these diagrams.

4.1.5 Use Cases

Use-Cases Describing the Software-System

Kruchten describes the RUP as a *„use-case-driven approach"* [Kruchten (2004), p. 30], meaning that use-cases are a thread through the whole development project linking the different disciplines and their resulting artifacts.

The employment of use-cases for describing the scope of a software system is not limited to the RUP. The concept goes back to Ivar Jacobson who described object-oriented software-development as a 'use-case-driven approach' and defined [Jacobson (1993), p. 159]:

> *„A use case is a specific way of using the system by performing some part of the functionality. Each use case constitutes a complete course of events initiated by an actor and it specifies the interaction that takes place between an actor and the system. A use case is thus a special sequence of related transactions performed by an actor and the system in a dialogue. The collected use cases specify all the existing ways of using the system."*

Independent from the RUP, use-cases are widely spread in software engineering, especially in the phase of requirements elicitation [Sommerville (2004), p. 154]. Therefore they could be one of the connections between software engineering in general and processes of organizational change.

For the purposes of this thesis it is important to notice that use-cases describe the technical system and not the organization. From the name 'use case' it could be implied that use cases are used to describe how the technical system is embedded into a context of use. It is well possible that such contextual information is contained in use-cases, but if so, then only as a side-effect. The purpose of use-cases within software engineering is clearly to define the technical system; the sum of all use-cases should completely describe the software's behavior and thereby *„define a firm boundary around the system"* [Kruchten (2004), p. 99]. Actions performed by members of an organization are of relevance if and only if they interact with the software.

Integration of Use-Cases into Self-descriptive Artifacts

Use-cases are suitable to serve as a supporting artifact in communication processes in-between users, because their content relates to spheres known to the users, and because their form is interpretable by users. Use-cases have no prescribed syntax; in fact they are often described in plain natural language ([Kruchten (2004), p. 100], [Rumbaugh, Jacobson, Booch (2005), p. 669], [Cockburn (2001), p. 1]). The UML does contain a type of diagram with specified syntax and semantic for the purpose of interrelating all use-cases in a project. These so-called use-case diagrams fulfill the task of describing the software's behavior completely. As a representation of the software's behavior in times when the software itself is not yet available, use-cases can support the process of structural coupling within the organization. They provide the organization with artifacts that precede the real change in the environment and that can be used to prepare organizational change ahead of time. Although the two case studies presented in chapters 6 and 7 did not employ use-cases, they did integrate early specifications of the software into diagrams representing work procedures. The experiences are discussed in section 7.2.1.

The Idea of having a Guiding Artifact throughout a CSCW-Project

The RUP includes the idea of utilizing use cases as guiding artifacts throughout a whole project. Different views on the software (such logical, implementation, process or deployment) are held together and validated by use cases selected for the use-case view [Kruchten (2004), pp. 87]. The idea of having one guiding artifact to which users and software developers can relate seems to be a good concept for integrating organizational change and software engineering; however use-cases as defined in the UML and used by the RUP fall short in supporting of structural coupling. They would have to be extended to include descriptions of the organization itself in order to belong to the self-descriptive artifacts. My methodological concept foresees socio-technical self-descriptions to be used as a guiding artifact throughout a complete CSCW-project. The methodological concept is presented in section 5.2.4; the experiences made during the case studies are discussed in section 7.4.

4.1.6 Business Modeling in the RUP

One of the nine disciplines of the RUP is called business modeling. Its purpose is to *„be sure that the applications we build help people in their daily chores and that they are not perceived as gadgets that do not offer added value."* [Kruchten (2004), p. 142]. Business modeling should precede the definition of use-cases and contains descriptions of the organization's purpose and goal as well as its organizational structures, its processes and its rules and regulations. At first sight, this concept links well with the needs of organizational change as it embeds use-cases and subsequent system design into a larger organizational context. But do the discipline of business modeling and the resulting artifacts actually support organizational change (cf. Term 12) as it is my intention for CSCW-projects (cf. Term 23)?

There are some problematic aspects in the way business modeling is described in the framework of the RUP. The first one is, that it remains unclear who the addressees of the artifacts of business modeling are. At one point it is implied that the software-developers can use them to document their understanding of the organization [Kruchten (2004), pp. 145]. But Kruchten also points out that business models *„are of concern to everyone involved in business development"* [Kruchten (2004), pp. 142]. The latter interpretation would be closer to the idea of using diagrammatic models as guiding artifacts throughout a CSCW-project. In this case however the RUP lacks methodological support, there is no guidance given about how the business models should be used in processes of organizational change.

Within UML it is suggested using activity diagrams to model business processes [Rumbaugh, Jacobson, Booch (2005), p. 160]. While this adds to the compatibility of business models with other models of the design process, it offers only insufficient support for the structural coupling of organizations and CSCW-systems. A large part of this thesis discusses the contents and forms of artifacts needed to support organizational change during a CSCW-project (cf. sections 5.2.1 and 5.2.2 for a presentation of the concept and sections 7.1 and 7.2 for a discussion of the empirical results); one of concepts that turns out to be essential is the integrated documentation of roles, activities and supporting IT-systems. This concept is not covered by UML's activity diagrams. Therefore my goals in this thesis exceed those of the business modeling in the RUP and UML.

One last problematic aspect shall be discussed. Kruchten introduces the role of a "*business designer*" [Kruchten (2004), p. 147]:

> *"The Business Designer details the specification of a part of the organization by describing one or several business use cases. He or she determines the business workers and business entities needed to realize a business use case, and also how they work together to achieve the realization. The Business Designer defines the responsibilities, operations, attributes, and relationships of one or several business workers and business entities."*

The role of a business designer who designs part of an organization just as one would design a software component contradicts the interpretation of organizations as autopoietic systems. An autopoietic system can only design itself; it cannot be designed by a business designer. Section 4.2 introduces systemic intervention as one approach of supporting an organization during change processes. Section 7.1.1 discusses the contents of socio-technical self-description as they emerged in the two case studies; these topics include the ones mentioned in the task description of the business designer. But the methods I suggest foreseeing that the discussion of these topics is part of the organization's structural coupling with the CSCW-system rather than the individual work of a "*business designer*".

It seems as if the inclusion of business modeling as a discipline into the RUP and as a model into the UML is an answer to the fact that software – especially if its purpose is to support cooperative processes – needs to be described in context of organizational processes. The RUP however does nothing but transferring methods that have proven their validity in the realm of software engineering to the realm of business development or organizational change. But the limits of methods provided by the discipline of software engineering seem to be reached at this point. Needed here are transdisciplinary approaches that link professional methods of organizational development with those of software-development.

4.1.7 Software in Use

Two aspects of software engineering relating to the use of software are discussed in this section: the deployment phase of the RUP and the design of operational procedures.

Deployment Discipline in the RUP

The deployment discipline comprises those activities that ensure the proper installation of the software at the customer's site. Within the RUP these activities include: Testing the software in its final operational environment, packaging the software for delivery, distributing the software, installing the software, training the end users and the sales force [Kruchten (2004), p. 237]. *„The purpose of the deployment is to turn the finished software product over to its users."* [ibid]. The fact that software engineering-processes such as the RUP foresee a deployment phase provides a link to processes of organizational change, because it shows that the usage of software is not taken for granted. Two statements about the RUP are cited for illustrating the importance of the deployment discipline for the usage of the software:

- The deployment discipline comprises *„the type of activities that take place when the software product is deployed in its operational environment and of the additional artifacts that may need to be developed for its successful use."* [ibid. p. 237].
- *„Successful deployment and indeed the overall success of the development effort are defined by the customer's willingness to use the new software."* [ibid. p. 239].

What is problematic about the deployment discipline as it is described in the RUP is that it does not acknowledge the social character of appropriation processes (cf. section 3.2.1, p. 83) and organizational change. The description of the RUP implies that by implementing the right features and providing the right artifacts (such as handbooks), the *„successful use"* can be planned or even guaranteed. There are no tasks foreseen for supporting the social process of structural coupling that might lead to a *„successful use"*.

Before discussing consequences for the methodological support, a related aspect of software engineering shall be introduced.

Design of Operational Procedures

Sommerville addresses the issue of use for software engineering in general. He defines socio-technical systems to include not only software but also operational processes into which the software is embedded (cf. section 2.1.3, p. 28; [Sommerville (2004), p. 21, 37]).

> *„Operational processes are the processes that are involved in using the system for its defined purpose. For example, operators of an air traffic control system follow specific processes when aircraft enter and leave airspace, when they have to change height or speed, when an emergency occurs and so on. For new systems, these operational processes have to be defined and documented during the system development process."*

But although Sommerville mentions the necessity of interdisciplinary teams [ibid. p. 26] for the design of socio-technical systems, he does not make any distinction regarding the methods to be applied. *„Designers should design operational processes to be flexible and adaptable"* [ibid. p. 38], just as software should be designed.

A similar argumentation as for the deployment discipline of the RUP applies here. The fact that the design of operational procedures is regarded as necessary and as closely related to software engineering provides a good link to processes of organizational change. But this links needs to be elaborated and methodologically supported.

The STEPS model for software engineering ([Floyd, Reisin, Schmidt (1989)], cf. section 4.3.3, p.121) suggests a division of labor between "developers" and "users". After a participative design, the developers are responsible for "software realization" while the users are responsible for "embedment preparation". Now, "embedment preparation" is neither identical with Summerville's operational procedures, nor with the RUP's deployment phase nor with *planning and establishing work procedures* from Figure 11, but it contains the idea that a process model can include a distribution of responsibility between the organization and a team of software engineers. I take this idea up for the methodological support: The process model for CSCW-projects as illustrated in Figure 11 differentiates the tasks of *planning and establishing work procedures* and *system design.* The final version of the process model (cf. Figure 59) will also include different teams which are responsible for these tasks.

The concept of socio-technical self-descriptions aims at supporting the organization in describing technically supported work procedures such that the social process of structural coupling is prepared and supported. This includes the description of operational processes as suggested by Sommerville. Section 7.1.1.1 discusses how work procedures were described in the case studies and section 7.1.1.4 discusses how agreements were made about the use of the CSCW-system by the work procedures.

4.1.8 Software Engineering reaching into Organizational Change

Although the definition provided in Term 6 which is taken up again at the beginning of this section (p. 91) excludes social processes of organizational change from the field of software engineering, there is a tight link between the two in the context of CSCW-systems. The examples of business process modeling (cf. section 4.1.6), the RUP's deployment discipline and the definition of operational procedures (cf. section 4.1.7) given above, show how software engineering reaches into the field of organizational change.

In the light of the special properties of organizations as social systems it is problematic that in the examples discussed above, software engineering methods are used to extend into the field of organizational change. By doing so, one tries to apply engineering methods to organizations.

It is not suggested here, that the discipline of software engineering should directly incorporate methods for supporting social processes of organizational change. But for the overall success of CSCW-projects it might be helpful to describe the establishment of technically supported work procedures as a social process which requires adequate methods which lie outside the scope of engineering methods. The interdependency between the two processes however requires that they are both included within any CSCW-project.

The concept of socio-technical self-descriptions attempts to support this approach: new artifacts representing the organization's socio-technical self-description are introduced for supporting technically triggered organizational change as a process of its own right. Then it is not the question of using methods from the field of software engineering for supporting organizational change but of relating methods and artifacts from the process of software engineering to those from the process of organizational change. Especially sections 5.2.4 and 7.4 discuss the aspects of integrating software engineering and organizational change.

4.2 Systemic Intervention

This section introduced systemic intervention as a consulting discipline which comprises methodological components realizing Luhmann's theoretical ideas (cf. section 2.3). Section 4.2.1 gives a brief introduction into the field of systemic intervention before section 4.2.2 provides an overview of published cases in which systemic intervention has been employed in IT-projects. Section 4.2.3 finally discusses central characteristics of systemic intervention and relates them to the methodological needs of CSCW-projects.

4.2.1 Towards an Understanding of „Systemic Intervention"

„Systemic intervention" is a consulting discipline that employs methods, which are based on the concepts of newer systems theory, for initiating and supporting organizational change. Although the term „intervention" itself is subject to debate among systemic consultants [Wimmer (1992), pp. 81], the following definition by Willke is widely adopted and I will adopt it for this thesis:

> **Intervention** is purposeful communication that respects the autonomy of the system that is intervened. A communication is considered purposeful, if it aims to achieve a certain effect.
>
> [Willke (1987), p. 333, translated from German by the author]
>
> An intervention is complete only after it brought about a functional equivalent for the previous situation.
>
> [ibid. p. 357]

Term 24: Intervention

If one uses systemic intervention as a concept for consultancy one has to cope with an inherent conflict: on the one hand there is the goal of achieving a *„certain effect"* within an organization; on the other there are the theoretical concepts suggesting that it is impossible for a consultant to change the behavior of an organization in a predictable way.

What does systemic intervention do [Wimmer (1992)]? Often the phrases „observing observations" or „2nd degree cybernetics" are used in this context. Systemic intervention tries to observe the way in which the organization observes itself and its environment. It is often applied when an organization is captured in rigid loops i.e. in repeating patterns of actions that are stable over a long period of time and that are perceived as problematic [Königswieser, Exner (2004), p. 18]. Systemic consultants watch and analyze the communications that take place within the organization; they search for the hidden and open rules that determine the flow of communications and that might be responsible for such dysfunctional loops. The consultants then support the organization in changing them. Systemic intervention as it is applied in organizations today has roots in systemic family therapy. This is true especially for those designs that are directed towards uncovering the hidden rules that are effective in an organization.

With this description systemic intervention seems to be very far away from software engineering; the next section discusses examples where methods of systemic intervention were applied in the context of IT-projects.

4.2.2 Reported Uses of Systemic Intervention in IT-Projects

Literature provides examples where systemic intervention is applied in the context of IT-projects. However in all examples the systemic intervention keeps at a distance from questions related directly to the development and adoption of IT-systems. Boos and Jarmai report a case where the deployment of SAP R3® was assumed to lead to structural changes. The consultants supported the client in restructuring teams and tasks. The SAP® system is mentioned several times like *„the new potentials of the SAP-system should consequently be used"* [Boos, Jarmai (2004), p. 135, translated from German]; or *„the goal could be reached, if the structure were improved significantly towards teamwork by introducing SAP R3® and a modified way of work."* [ibid. p. 136]; and finally *„For the development of these structural suggestions the following basic conditions were taken into account: … deployment of SAP R3® and concentration on central tasks."*. The authors treat the IT-system and its adoption as a solid black box; they address neither options nor necessities for a detailed discussion of the work-procedures associated with SAP R3®.

In [Kuhnt (1998), pp. 115][20] four case studies are reported in which the processes of software engineering were supplemented by systemic interventions. In the first case study, systemic intervention was initially used for a context analysis identifying interests, goals, success factors and fears associated with the project; then intervention designs were employed for supporting the expansion of creative potential during the elaboration of business processes. The second case study is labeled "evaluation and establishment of tools and methods" and seems to be close to my focus in this thesis [ibid. p. 152]. But in the course of the intervention, its focus moved [ibid. p. 160]: away from the methods

[20] The main results from her thesis [Kuhnt (1998)] were also published in [Kuhnt (1997)]

and tools towards communication and cooperation between planners and developers. The third case study [ibid. p. 161] was about quality assurance within an "informatics support center" and was not related at all to the design and deployment of a software systems. The fourth case study [ibid. p. 167] is concerned with a specific software system. In the end however, interviews show that the software is not used, and as a consequence of the intervention the group which had developed the software was closed down. While all four case studies provide helpful insight into processes occurring within a design team and in-between design team and organization, the question how an organization can be support in its structural coupling with a CSCW-system is only covered marginally.

The systemic approach described in [Huber, Kuhnt (1995)] bases on the systemic concepts described in section 2.3 and suggests combining them with an evolutionary approach to SWE. With each iteration of the software, the organization enters a phase of learning and the insights lead to the next level. System theory is used to explain why the way the organization reacts to the software is not predictable; but the issue how an organization can actively prepare for the adoption of an IT-system is not addressed.

The resume at this point is that, according to current literature, methods of systemic intervention are used in the context of IT-projects but that they do not really touch issues of structural coupling between an organization and a CSCW-system. This is not to say that the issues covered by interventions described in the case studies are not important; the point is that they could go one step further to address issues directly related to a software-system. The following sections discuss how central principles of systemic intervention relate to CSCW-projects.

4.2.3 Systemic Intervention and CSCW-Projects

Respect for the Autopoietic Nature of Organizations

In [Wimmer (1992), pp. 59] it is said that systemic interventions describe more an attitude towards social systems, than procedures for intervention. The most important aspect of the attitude behind systemic interventions is the respect for the autopoietic nature (cf. 2.3.2) of social systems. Their self-referential closure is essential for their survival and any intervention must have the goal to preserve it [Willke (1994), p. 148]. If software engineers ceased to view an organization, its structures and processes as something to be „designed" just as software is designed (cf. 4.1.7), they might be better prepared for the unpredictable courses that appropriation of a CSCW-system by an organization may take.

If an organization is interpreted as an autopoietic system, then it follows that the incorporation of a CSCW-system into work procedures cannot be the result of a design by software engineers but rather the result of a process of structural coupling that needs to take place within the organization itself. It is an important principle of systemic intervention that the responsibility for actually changing remains with the organization [Willke (1994), p. 206].

The integration of users into the design processes of software has a long tradition and there are quite different motivations for doing so. There is the pragmatical argument that users are an important source of information for specifying and validating the software (cf. section 4.1); but there are also humanistic or political arguments that people should have the right of influencing the design of the tools with which they work (cf. section 4.3.3, p.113). In the context of systemic intervention, yet another argument for user participation arises. It has been said that the outcome of a CSCW-project is twofold: there is the technical CSCW-system and there are new work procedures that incorporate the technical CSCW-system. If the *planning and establishing of work procedures* (cf. Figure 11) is regarded as a social process that is to be performed by the organization itself, then users need to participate in CSCW-projects because they are the only ones that can actually incorporate the CSCW-system into their work procedures. With this perspective the users' task is not only to support the design of an adequate CSCW-application, they have the additional task of *planning and establishing work procedures* that make use of this CSCW-application.

These insights have consequences for the organization, more precisely the staffing, of a CSCW-project: the workshop design STWT foresees the direct participation of users (cf. section 5.2.3); the process model illustrated in Figure 11 documents *planning and establishing work procedures* as a separate task, and section 8.2 suggests that CSCW-projects are organized such that the members of the organization bear the responsibility for this task.

Self-Descriptions and Descriptions by Others

It has been said that (socio-technical) self-descriptions serve two purposes within the context of organizational change: first, the explication of self-descriptions can be used as the basis for reflection and subsequent change. Methods of systemic intervention make use of this function [Wollnik (1998)]. Second, socio-technical self-descriptions help stabilizing an organization by serving as guidance (cf. section 2.4.4).

However in order to support organizational change processes, the term self-description needs to be taken literally: the system must describe itself. It does not help to pay a consultant (or software engineer) to describe the flow of communications within the system and then make suggestions for change. There is a problem for the integration of software engineering and systemic methods connected to this point: from a systemic point of view descriptions by others are not only useless but potentially harmful [Wimmer (1992), p. 79 and p. 100]. Consultancies as well as software engineering projects usually start with a phase to analyze the status quo. Methods employed comprise interviews, observations or analysis of documents. The goal is that the consultant or software engineer learns about the organization such that she is knows enough about the field to conduct the project. One problem is that these seemingly neutral actions for collecting information about the system are not without effect on the system. Expectations as well as fears concerning an external „objective" and „comprehensive" feedback are raised. Another problem is that neither diagnosis nor suggestions derived from it are connectable within the organization unless they are brought within its communicative context.

Systemic intervention copes with these problems by designing any diagnosis as an intervention (i.e. Designs A1 – A6 in [Königswieser, Exner (2004), pp. 164]). i.e. not the consultant diagnoses the organization but the consultant supports the organization in diagnosing itself. Any mismatch between the organization's self-perception and the consultant's perception may but does not need to be subject to debate. Such an approach is possible because the systemic consultant does not have the need to gain a full picture of the organization that is agreed by the organization. A consultant may even do without a thorough inquiry phase at the beginning, if she thinks that this is appropriate.

Software engineers have different needs: a stable description of the organization that is agreed within the organization and between the organization and the software engineers is the basis for their work. Other than the systemic consultant, a software engineer needs to gain an understanding of the organization that enables her to code part of its procedures into software. Therefore the diagnosis phase within a CSCW-project cannot be limited to helping the organization describe itself but must be augmented such that the resulting description can be used by the software engineers to continue the technical design and by the organization to continue its process of organizational change.

There is no perfect solution to this inherent conflict. The first step is to create awareness for the problem - within a CSCW-project there needs to be room for both aspects. The methodological support for socio-technical self-descriptions includes three elements relating to this issue: interdisciplinary workshops in which software engineers and members of the organization jointly discuss current and future work-procedures (cf. section 5.2.3); semi-structured diagrams that can serve as boundary objects between the users and software engineers (cf. section 5.2.2); a consultant / facilitator who is not part of the software engineering team and is therefore not caught in his own needs to gather information (5.2.3).

Individual and Social Level in CSCW-Projects

Just as system theory is a social theory that explains social phenomena; systemic intervention addresses social issues in organizations (rather than individual problems such as personal motivation). From the fact that social systems are viewed as consisting of a network of communications it follows that organizational change means changing this network by changing the rules on which it is based. Systemic intervention attempts to make these rules visible and support change if appropriate.

This focus on the rules within an organization matches the needs of CSCW-projects in two respects. First, it relates to the task of requirements engineering. Since CSCW-systems include inscriptions of organizational processes, it is helpful to clearly identify the rules underlying these organizational processes. Second, the planning of the usage of the CSCW-system triggers a process of structural coupling which also needs to take place on the social level (cf. section 2.3.5 and 2.3.7).

Although it is a central aspect of systemic intervention to look beyond the individual, the individual is not ignored: Luhmann derives the concept of interpenetration between psychic and social systems (cf. section 2.3.5); Willke cites Senge [(1990), p. 139]: *„Organizations learn only through individuals who learn. Individual learning does not guarantee organizational learning. But without it no organizational learning occurs."* Within CSCW-projects, the interdependency between individual and social level manifests itself in the topic of training. A CSCW-project that only provides training and help for individuals with the goal that they understand the functionality of the CSCW-system remains on the individual level and does not support the social level. But the training on the individual level is a prerequisite for the individual member to be able to participate in a process of organizational change. Only those who understand the functionality of a CSCW-system can discuss rules that embed the software functions into the organization's communications. Both case studies include qualification workshops in which individual learning was combined with the discussion of additional conventions needed to incorporate the CSCW-application into the work procedures (cf. Example 34).

4.3 Design Methods in the Field of CSCW

4.3.1 Introduction

The proximity of social issues and software engineering has inspired many methods in the field of CSCW that address the integration of the two. The purpose of this section is threefold:

1. Collect best practices provided by these methods to show the state of the art behind which no new method should fall back
2. Indicate those areas where existing methods fall short in actively supporting the structural coupling between organization and CSCW-system.
3. Position the theoretical concept of socio-technical self-descriptions as well as its methodological implications within the context of existing research and methods in the field of CSCW.

The relevant methods presented were selected according to the following criteria:

a) Methods that include means for integrating technical design and organizational change are considered.

b) Methods that are prominent within the field of CSCW are considered, in order to be able to position my thesis within the context of other research.

Discussing the selected methods brings a systematic problem about: most of them are so broad that it is not definite what one can or cannot do with a method. All the methods described in the following sections resulted from research of many years and have advanced the field of CSCW-research. Pointing out their strength and weaknesses with regard to my goals is not easily possible because in many cases one could think of a way in which the method could be modified or adapted to suit the needs. The following sections focus on the original intents and published uses of the methods, section 8.3 discusses possible extensions and combinations with my concepts.

4.3.2 Methodological Impact of Tavistock Institute

The methodological concretion of the Tavistock's „socio-technical approach" comprises a set of principles and related procedures for analyzing organizations as socio-technical systems. Both are summarized in Table 3 and Table 4:

Five Principles of the socio-technical approach

1) The concept of the socio-technical system: the need to study a production system as a whole and to understand the interrelatedness of all its aspects.
2) The organization as an open system: The organization must be seen as an open system in constant interaction with its environment. This interaction must be taken into account in the analysis.
3) The principle of organizational choice: the importance of matching the social and the technical systems together in the most appropriate way. "joint optimization of the socio-technical system".
4) Autonomous groups lead to best results:
   - Degree of responsibility for a major section of the task
   - Own setting of targets
   - Management of its own internal relationships
5) Alienation from work: the insight that poor morale and lack of motivation were result of the type of work role created by the use of technical systems.

Table 3: Five Principles of the Socio-technical Approach According to [Hill (1971) p. 32]

Nine-Step Model of Socio-technical Analysis

Step 1: Initial Scanning
Step 2: Identification of Unit Operations
Step 3: Identification of Key Process Variances and their Interrelationships
Step 4: Analysis of the Social System
Step 5: People's Perceptions of their Roles
Step 6: Maintenance System
Step 7: Supply and User System
Step 8: Work Environment and Development Plans
Step 9: Proposals for Change

Table 4: Nine-Step Model of Socio-technical Analysis according to [Emery (1993)] [21]

The central aspect of Tavistock's methods for organizing industrial work is the autonomous work group; they initially derived their concepts from the observation of coal miners successfully organizing their work in such groups even though modern technology might suggest a more tayloristic organization (cf. section 2.1.2). The idea semi-of autonomous work groups as well as the principles became influential in the organization of industrial work and were further developed by other researchers (e.g. [Cherns (1987)], [Udris, Ulich (1987)]).

For the field of CSCW, the suggested principles and methods need to be transferred before they can be applied. Some of the methods that are introduced in the next section include direct relations to the Tavistock's research and transfer them to IS development (ETHICS, p. 110; Eason's socio-technical design, p. 112; Scandinavian participatory design, p. 113; MUST, p. 115). Under the title "The New SocioTech" the socio-technical approach is persued more in the sense of a paradigmatic idea than in the sense of methodological support [Coakes, Willis, Lloyd-Jones (2000)].

[21] Emery presented these nine steps at meeting in 1967; others such as [Sydow (1985)] and [Hill (1971)] also refer to these nine steps.

For the methodological support of CSCW-projects by using socio-technical self-descriptions, two aspects from the socio-technical principles (cf. Table 3) are of central importance. First, the notion of "organizational choice": the researchers from the Tavistock found that no technology completely determines the work organization, there is always an organizational choice how to design work settings that match a given technology. For CSCW-projects this implies a degree of freedom as well as a task, because the work organization matching a new CSCW-system needs to be planned. Other than in the Tavistock's studies, where the machine was fix, the CSCW-system itself is subject to design within CSCW-projects. The process model which is suggested for CSCW-projects in 3.2.2 (Figure 11) respects these insights by including the two interrelated tasks of *Planning and establishing work procedures* and *Technical design and implementation of software.*

Second, the notion of "joint optimization": the researchers from the Tavistock emphasize that the socio-technical system as a whole can only be optimal, if both of its components are optimized in relation to each other. I included this concept of joint optimization by starting the process of *Planning and establishing work procedures* in parallel with *Technical design and implementation of software* (cf. section 3.2.2, p. 87).

4.3.3 Overview of Methods in the Field of CSCW

ETHICS, transferring socio-technical Concepts to IT-Design

The acronym ETHICS stands for *„Effective Technical and Human Implementation of Computer-based Systems"* [Mumford (1995), p. 27] and denotes a *„systems design methodology"* [ibid. p. 3] that Mumford elaborated on the basis of the socio-technical theory developed in London's Tavistock Institute [Trist, Bamforth (1951)]. ETHICS was selected as a relevant method for two reasons. First, Mumford refers to the theory of socio-technical systems explicitly [Mumford (1996), pp. 64] and thereby shares historical foundations with my approach. Second, she defines that *„Socio-technical design is an approach that aims to give equal weight to social and technical issues when new work systems are being designed."* [Mumford (2000), p. 125]. This definition raises the expectation that ETHICS provides methodological support for the three subtasks of a *CSCW-Project* as they are illustrated in Figure 11.

Mumford [(1995), p. 27] lists three general goals for ETHICS:

„

1. *To enable the future users of a new system to play a major role in its design and to assume responsibility for designing the work structure that surrounds the technology …*
2. *To ensure that new systems are acceptable to users because they both increase user efficiency and job satisfaction.*
3. *To assist users to become increasingly competent in the management of their own organizational change so that this becomes a shared activity with the technical specialists and reduces the demand for scarce technical resources."*

Table 5 summarizes methods and procedure provided by ETHICS for these purposes.

Steps in ETHICS	**Methods for Diagnosis and Design in ETHICS**
Diagnosing user needs and problems, focusing on short- and long-term efficiency, job satisfaction and quality.	A framework to assist the identification of mission, key tasks, important constraints and factors critical to effective operation
Setting efficiency, effectiveness, job satisfaction and quality objectives.	A variance analysis tool to assist the identification of significant problems and problem areas.
Developing a number of alternative design strategies which will assist the chosen efficiency, effectiveness, job satisfaction and quality objectives.	A questionnaire to measure job satisfaction
Choosing the strategy which best achieves all of these objectives.	A framework to identify what is likely to change in the internal and external environments.
Choosing hardware and software, and designing the system in detail.	A set of guidelines for individual and group work design.
Implementing the new system.	
Evaluating its success once it is operational.	

Table 5: Steps and Methods defined in ETHICS

[Mumford (1995), pp. 28]

The focus of ETHICS is on developing and using psychological instruments such as questionnaires for measuring job satisfaction or a variance analysis tool for assisting the identification of significant problems in the context of IT-projects. The results of such inquiries are used in the process of requirements elicitation and in the evaluation of different design approaches. Mumford‘s research is formed by the QWL-tradition (Quality of Working Live). Her goal is to shape work conditions in such a way that they are in line with a humanistic idea of man. And

> *„ETHICS … seeks to legitimize a value position in which the future users of computer systems at all organizational levels play a major part in the design of these systems. The argument here is that people should be able to influence the design of their own work situations …"*

[Mumford (1995), p. 3]. Part of her vision of working live is a very harmonic cooperation between management and workers. A management that cares about the workers' needs and allows them freedom in shaping their workplace will create job satisfaction [Mumford (1996)].

The discussion in section 4.4 will come to the conclusion that ETHICS remains very abstract in describing how the results of the socio-technical analysis should be included in system design and that it does not provide methodological support for the task of structural coupling. However, Mumford contributed to the research on software design methods by drawing the line to the socio-technical research of the Tavistock Institute; she thereby opened a reservoir of research (e.g. [Cummings, Srivastva (1977)]) that stems from traditional industry such as coal-mining and textile, but that nevertheless induced the basic insights about the interdependence of technical and social systems (cf. sections 2.1.2, 4.3.2).

Eason's socio-technical Design of Information Technology Systems

Just as Mumford, Eason derives his concepts for IS design by referring to insights from the early Tavistock research (e.g. [Eason (1988), p. 45]); but in comparison to ETHICS, Eason is more precise concerning the integration with methods of IS design. The core of his methodology is a framework comprising 10 propositions for the design of socio-technical information systems (cf. Table 6). While these propositions are quite abstract, their descriptions are more concrete and include methodological support

10 Propositions for Socio-Technical Design of Information Systems

Proposition 1: The successful exploitation of information technology depends upon the ability and willingness of the employees of an organization to use the appropriate technology to engage in worthwhile tasks.

Proposition 2: The design target must be to create a socio-technical system capable of serving organizational goals, not to create a technical system capable of delivering a technical service.

Proposition 3: The effective exploitation of socio-technical systems depends upon the adoption of a planned process of change that meets the needs of people who are coping with major changes in their working lives.

Proposition 4: The design of effective socio-technical systems will depend upon the participation of all relevant "stakeholder" in the design process.

Proposition 5: Major benefits will only result if the socio-technical developments are directed at major organizational purposes where there are opportunities to be taken or problems to be resolved.

Proposition 6: The specification for a new socio-technical system must include the definition of a social system which enables people in work roles to co-operate effectively in seeking organizational purposes and provides jobs which incumbents perceive as worthwhile.

Proposition 7: Information technology systems must be designed to serve the functional needs of the organization by serving the functional needs of individual users in a usable and acceptable way.

Proposition 8: The effective exploitation of information technology requires a major form of organizational and individual learning.

Proposition 9: The exploitation of the capabilities of information technology can only be achieved by a progressive, planned form of evolutionary growth.

Proposition 10: To be successful, socio-technical design concepts must as far as possible complement existing design procedures and organization change practices.

Table 6: Eason's 10 Propositions for Socio-Technical Design of Information Systems (cf. [Eason (1988), pp. 44)

Based on these 10 propositions, Eason describes and discusses methodological components suitable for conducting a complete project of IT development integrated with organizational development and job design. His goal is not the elaboration of a closed method for IT projects, he rather aims at providing a toolbox [ibid. p. 51] from which methods can be used as appropriate for a specific setting. However Eason describes two "*overriding features of the design process which have to be present in some degree if it is to be possible to introduce these ideas*" [ibid. p. 50]: the first is user participation and the second the "*recognition that change is an evolutionary process*" [ibid.]. Since these prepositions are not contradictory to my theoretical concept of socio-technical settings, the methodological support I elaborate for socio-technical self-descriptions (cf. section 8.2) could be included in the toolbox for projects following Eason's propositions. The support for socio-technical self-descriptions would add an important aspect that is missing in Eason's concepts and consequently in his toolbox: the observation that the adoption of an IT-system is again a social process that needs to be supported. Eason includes methods for organizational design and job design [ibid. pp. 52] and he links them to the design of technical features, but social phenomena such as structural coupling between organization and technical system (cf. section 3.2.1, p. 83) are not included.

There is one interesting detail that shows how Eason handles the interrelation between organization and IS, and that also bears relations to the methodological support I elaborate. The topic is concerned with prototyping as part of the system specification process [ibid. pp. 101]. Here Eason talks about the problem that *"the capability for rapid prototyping can cause designers to rush into constructing a technical solution forgetting that what is needed is a socio-technical solution."* [ibid. p. 101]. As a remedy he suggests specifying the division of labor between social system and technical system prior to prototyping, and he further requests that this allocation is documented in form of diagrams showing work procedures as well as processing steps in the technical system.

The methodological support for socio-technical self-descriptions includes diagrams that contain the information requested by Eason, but they also exceed his idea by serving as a guiding artifact throughout the whole project (cf. section 5.1.2). They do not only document a division of labor at one point of the project, the diagrams documenting socio-technical self-descriptions will be used throughout the project helping the project team to keep a socio-technical perspective.

Participatory Design (PD)

The Scandinavian school of participatory design shares some roots with Mumford‘s ETHICS: socio-technical concepts developed at London's Tavistock Institute found their way to Scandinavia through cooperation with the institute for „Industrial Social Research" in Trondheim, Norway beginning in 1962 [Hill (1971), p. 34]. The Norwegian „Industrial Democracy Project" [Emery, Thorsrud (1976)] was a joint venture of the state of Norway, the unions and industry with the goal to increase productivity by using the scarce human resources as sensibly as possible. The unions wanted to put through their request for more participation in order to reduce the workers' alienation from their work. Originating from these roots participatory design has established itself as an influential stream of research with close connections to CSCW, that can be defined as follows [Webpage CPSR]:

> *„Participatory Design (PD) is an approach to the assessment, design, and development of technological and organizational systems that places a premium on the active involvement of workplace practitioners (usually potential or current users of the system) in design and decision-making processes."*

There are close relations between PD and the field of CSCW, because user participation is an important element in CSCW-projects; although unlike PD, the field of CSCW considers user participation as a means and not a goal [Kensing, Blomberg (1998), p. 181]. CSCW-projects rely on intensive workplace studies for informing the design of CSCW-applications and PD techniques can be employed for this purpose, [ibid. p. 182]. In section 4.2.3 (p. 105) yet another argument was given for user participation in CSCW-projects: if organizational change processes are interpreted as autopoietic processes, they can only be performed within the organization itself; hence, the participation of future users in a CSCW-project needs to be supported.

PD in general concentrates on methods to integrate people who are no software-experts into the design process. Therefore PD tends to avoid any formalism and any software engineering methods that could discourage users. In [Ehn, Sjögren (1991)] the authors describe how they would organize a Future Workshop in a Locomotive Repair Shop with the title „Computer Support for material administration?". Within this workshop games would be played to encourage the participants to think (even fantasize) about their future work. The authors conclude [ibid. p. 267]:

> *„In the design process that follows this Future Game, more games may be played, and mock-ups and/or prototypes experienced. There will certainly also be a time and a place for design artifacts such as detailed technical system descriptions, but that is another story. The story we have been telling about scripts for action is about the participatory side of design, and the necessity of taking users' experience seriously. That is why we have been playing games, not at the price of seriousness, but as a necessary precondition for engaged and more democratic participation."*

The cited example of future games is typical for PD methods which often describe workshop designs; [Kensing, Munk-Madsen (1993), p.81] or [Muller, Wildman, White (1993), p.27] provide overviews. The methods I elaborate include workshop designs which are inspired by PD-techniques (cf. section 5.2.3), although no single PD-technique is adopted completely. But the underlying values such as respect for the users' expertise in their own work and the necessity for workshop designs that allow users who are no technical experts to participate in a design process have important impact on the methodological concepts.

PD in general is not limited to the design of IT-systems [Kensing, Munk-Madsen (1993), p. 181], but in fact many publications report on techniques for organizing user participation in designing information technology (e.g. [Greenbaum, Kyng (1991)]). Most of the techniques concentrate on including the users' experience in the design of computer systems; fewer examples illustrate how users can participate in planning work procedures. One of these examples is found in [Ehn, Sjögren (1991), pp. 256]. It shall be cited briefly, because it also touches the issue of commitments and conventions (cf. section 3.1.2, p. 68). The authors describe an organization that had invested in desktop publishing technology without reaching the goals they had hoped for. The organization now needed to develop professional roles and to restructure the work organization in order to make use of the technology and in order to improve the typographic quality. In a workshop conducted by the authors, the participants played games in which they discussed situations that could occur during their work. One example for such a situation was that the toner cartridge is suddenly empty while a time-critical print job is still printing. The group defined a commitment for a role [ibid]: *„In my role as a publication administrator I commit myself to solve the immediate problem by ordering a new cartridge and have it delivered by taxi. In the long run I will establish a routine for the maintenance of the laser printer. My condition is that I get appropriate training in maintenance of the printer."*

In terms of system theory this is an example of structural coupling by the organization; the self-description is extended by describing a new task for a role which is necessary for operating the technical system.

Both of my case studies confirmed that commitments like the one in the example are necessary for integrating it into work procedures; but they also showed that it is possible to start the discussion of such agreements before the CSCW-system is put into use. Therefore the contents of socio-technical self-descriptions foresee „agreements concerning the usage of the CSCW-system" as a separate category (cf. section 7.1.1.4). Exceeding the mentioned PD-techniques however, it is suggested documenting such commitments in form of semi-structured diagrams which integrate technical and organizational descriptions (cf. sections 5.2.2, 7.2, 8.2)

MUST, a Method for Participatory Design (PD)

MUST is a Danish acronym for *„Theories of and methods for design activities"*; in [Kensing, Simonsen, Bødker (1998)] the authors describe MUST as *„a method for participatory design"*.

MUST is singled out from the PD techniques described in the previous section, because it exceeds the level of workshop designs and addresses issues of project organization. The methods I elaborate share an important basic assumption with MUST; namely the principle of *„co-development of IT, work organization and user qualification"* [Kensing, Simonsen, and Bødker (1998), p. 183].

MUST as a method for participatory design has a long history and its components have been proven by an extensive amount of empirical research. Kensing, Simonsen, and Bødker [(1998), p. 169] describe MUST as a *„conceptual framework of the design process … [it] focuses on the early activities in a development process."* MUST contains five activities and six principles for participatory design which are summarized in Table 7 [ibid. pp. 174 and pp. 183].

Five Activities of MUST	Project establishment; Strategic analysis; In-depth analysis of selected work domains; Developing visions of the overall change; Anchoring the visions.
Six Principles of MUST	Participation; Close links to project management; Design as a communication process; Combining ethnography and intervention; Co-development of IT, work-organization and users' qualification; Sustainability.

Table 7: Activities and Principles of MUST

The authors restrict the applicability of MUST to a design phase which precedes a "contractual bid and selection". The phases of "delivery and / or development" and "use and design-in-use" are not meant to be supported by MUST anymore [ibid. p. 172]. The methodological components for supporting socio-technical self-description are explicitly not limited to certain phases of a CSCW-project; nevertheless since the underlying assumptions of MUST and my approach are so close, socio-technical self-descriptions could be included into a MUST-project. Maybe it would even be possible to extend the applicability of MUST (cf. section 8.3.2 for a discussion).

Joint Application Design (JAD)

Joint Application Design (JAD) is a workshop-based method that was designed in the 1970s at IBM to support the phase of requirements engineering in technical projects [Crawford (1994), p. 1]. JAD is

> *„Teamwork involving business and technical people to identify business objectives and define design requirements and operational specifications for the application of nontechnical and technological solutions to business problems."*

The fact that JAD addresses business problems by combining nontechnical and technological solutions makes it relevant in this context. Andy Crawford who elaborated the JAD renounces any scientific background explicitly by stating that he works *„without an academic environment and backing for theoretical research to other written works"* [Crawford (1994), p. xiii]. Nevertheless JAD is cited and discussed in academic publications such as [Coughlan, Mcredie (2002)].

JAD is an American method that aims at making the design process more efficient in terms of time and money. Crawford himself points out the amount of savings his method brings in comparison to other design methods at several places in his book. The participation of users in the design process is a means for this goal. Such the motivations for participatory components in JAD are quite different from those in the Scandinavian approach (cf. p. 113 in this section) but nevertheless both seek to let users actively participate in the design process. For a detailed comparison of PD and JAD refer to [Carmel, Whitaker, George (1993)].

At the core of JAD there are workshops in which technical and non-technical people work together. Crawford argues that workshops are a more efficient method to bring in the perspective of all stakeholders than interviews with each of them. Also, they are preferable because they support the communication between the stakeholders. JAD includes a detailed description of the participating roles and the sequence of workshops.

In this respect JAD adds to the workshop designs suggested by PD (cf. p. 113 in this section) and has generally influenced the workshop design I suggest (cf. section 5.2.3). But with regard to systematic documentation JAD exceeds the PD-techniques: The so-called „workbooks" which contain all agreements made during a project are an important part of the method JAD. Crawford describes the systematic structure of the workbooks as far as including dividers between sections and suggestions for forms in detail. All participants have copies of the workbook and relevant pages are projected during the workshops. The theoretical background discussed in chapters 2 and 3 implies that documenting socio-technical self-descriptions can support the process of structural coupling between organization and CSCW-system. Therefore the creation and maintenance of documents containing the organization's socio-technical self-description are an important part of my methodological concepts (cf. sections 5.2.1 and 5.2.2). JAD workshops are a good example how documents can lead through a series of workshops; the concepts of StSd however differ from JAD by using graphical models and keeping closer to the technical design process (cf. discussion in section 4.4).

Contextual Design (CD)

> *„Contextual design is a full front-end design process that takes a cross-functional team from collecting data about users in the field, through interpretation and consolidation of that data, to the design of product concepts and a tested product structure."*

[Holtzblatt (2003), p. 942]. At first glance, CD does not seem to relate to the topics of this thesis at all. CD is a method for designing software-products for a big market, not for an individual organization. It therefore does not relate issues of software engineering to issues of organizational change in the sense of integrating them in one CSCW-project. There are three reasons why CD is considered relevant nevertheless. First, parts of CD such as contextual inquiry (CI) are proven methods for IT-projects for collecting data about work practice and are thus relevant for CSCW-projects [Holtzblatt, Beyer (1996)]; second CD makes extensive use of graphical artifacts representing work practice and can thus inform the usage of graphical artifacts as socio-technical self-descriptions (cf. section 5.2.2); third CD is interwoven in CSCW-related research by mutual citation (see below).

Holtzblatt describes the situation of software industry in the mid-1980s and the problem to which CD gives the answer [Holtzblatt (2003), p.942]:

> *„The problem was that the data needed for design was simply not being collected. To make products more usable, to make products that people really wanted and could use meant understanding what people were really trying to do and designing new technology to support, extend, and transform that practice. But without knowing the practice, missing the target or breaking the practice was probable."*

At the beginning of CD was contextual inquiry (CI) [Beyer, Holtzblatt (1998), p. 20]: a method to collect data about people's work practice in order to make use of it in the design of software products. Over the years CD was developed and refined. Today CD consists of six steps and corresponding models that are summarized in *Table 8*.

Step	**Description**
Contextual inquiry	Collect data using ethnographic techniques by observing and questioning customers while they work.
Interpretation Session and Work Modeling	Capture the key issues of one individual's work practice and model their work using the five work models, each capturing a different dimension of work practice.
Affinity and Work Model Consolidation	Consolidate the individual work models and issues to reveal the structure of the work across a population without losing individual variation.
Visioning and Storyboarding	Invent how new technology will address the user work practice by creating a high-level story (the vision) of how work will be changed. Work out the details of this story at the level of the task (storyboard).
User Environment Design	Design a floor plan that shows how the parts of the new system will interrelate. Represent the structure and function clustering of the system independently of consideration of User Interface and implementation.
Paper Prototyping	Test and modify the new system design in partnership with customers using paper mock-ups of the User Interface. Have people do real work tasks in the paper system, not just review it and provide opinions.

Table 8: Descriptions of Steps in CD

[Holtzblatt (2003), p. 943]

It is important to note that CD aims at informing technical designers and programmers about work practices such that they are able to design software that matches these work practices. CD supports designers of software-products rather than designers in customer-specific software projects. The goal of CD is not to involve users in designing their own technical support. Nevertheless its solid methods for collecting data from field work made CD and CI also relevant for some researchers interested in participatory design (cp. [Simonsen, Kensing (1997)] or [Muller, Wildman, White (1993)]). And CD on the other hand draws on participatory methods such us mock-ups to ensure that users are able to *„co-design"* [Beyer, Holtzblatt (1998), p. 371].

Soft Systems Methodology (SSM)

When Checkland published his book *„Systems Thinking, Systems Practice"* and with it his Soft Systems Methodology" (SSM) in 1981 [Checkland (1999)][22], he thought neither of designing IT-systems nor of planning their use. His point was to describe systemic paradigms for organizational change independent from any technical systems that might be in use. In the years following the initial publication, Checkland tried out and improved his methodology in numerous case studies. In the meantime the relevance of the organizations' IT-infrastructure with respect to work-organization and problem solving grew and issues of IT-Design were added to SSM. The links between IT-infrastructure and SSM are documented in [Checkland, Scholes (1990)] and [Checkland, Holwell (1998)]. SSM is included here for two reasons. First, its theoretical background in systems theory (cf. discussion in section 3.1), and second its broad reception within the field of CSCW.

SSM suggests four steps to be taken in a project which are listed in *Table 9*.

Four Steps in SSM [Checkland (1999), p. A15]

1. Finding out about a problem situation, including culturally / politically;
2. Formulating some relevant purposeful activity models;
3. Debating the situation, using the models, seeking from that debate both
    a. Changes which would improve the situation and are regarded as both desirable and (culturally) feasible, *and*
    b. The accommodations between conflicting interests which will enable action-to-improve to be taken;
4. Taking action in the situation to bring about improvement.

Table 9: Four Steps of SSM

With regard to IT-systems SSM acts in three steps: Since within the conceptualization of SSM any technical system is designed for supporting a 'purposeful activity system', the first step needs to identify the *„purposeful activity system"* together with its goals; then information flows and technical systems supporting them can be defined [Checkland, Scholes (1990), p. 270].

Integral parts of SSM are the so-called „rich pictures" [Checkland (1999), p. A21]. Rich pictures are sketches that describe a problematic situation within an organization graphically. According to Checkland the only purpose of rich pictures is to enable and support communicative processes within a group. They are not meant to formally document an information system. Within the fields of PD and CSCW rich pictures gained certain popularity because the combine the advantages of having graphical artifacts while avoiding the pitfalls of formal representations of work (cp. Section 4.3.5). The idea of using rich pictures has been adopted by other methodological frameworks such as MUST (cf. p. 115) or Multiview2 (cf. p. 119).

[22] The book referenced here was published in 1999. It contains a complete, unchanged edition of the original publication from 1981 supplemented by an introductory chapter "A Thirty Year Retrospective".

My own methodological components are influenced by SSM insofar as SSM is yet another example of a method supporting IT-related communication processes within a group with diagrams; this influence becomes evident in the workshop-design (cf. section 5.2.3) as well as in the form of documenting socio-technical self-descriptions (cf. section 5.2.2). The latter takes up the idea of avoiding formalisms which are difficult to understand; however it will exceed rich pictures by introducing semi-structured diagrams which do intend to document aspects of the CSCW-system as well as conventions related to its use.

Multiview2

> *„The Multiview methodology was proposed as a framework for information systems development that would take account of the complex world of people and organizations as well as information and communication technologies."*

[Avison et al. (1998), p. 124].

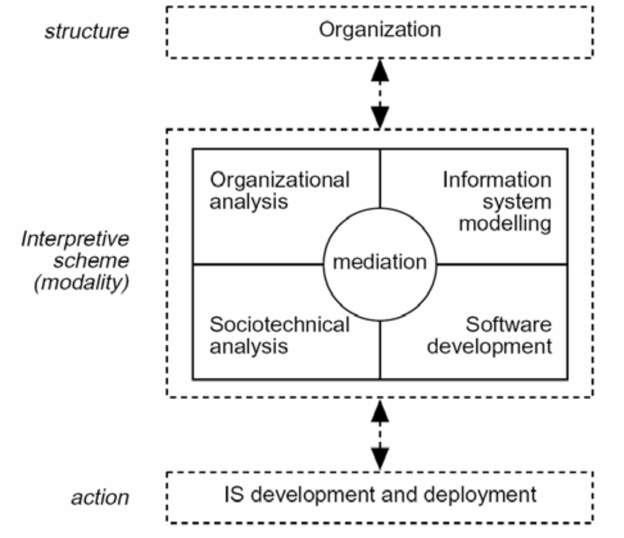


Figure 13: Framework of Multiview2

[Avison et al. (1998), p. 130]

I consider Multiview2 (which is a redefinition of Multiview [Avison, Wood-Harper (1990)]) as relevant for two reasons. First, it addresses the integration of organizational and technical issues; second, the framework Multiview2 refers to several of the methods discussed in this section

Multiview defines four thematic areas in which information-system-development must take place: organizational analysis, information system modeling, socio-technical analysis and software development (Figure 13). The authors emphasize that these four components are interdependent and cannot be put into a sequential order.

Multiview2 is a methodology on a meta-level that aims to describe the essential components of IT-projects without prescribing methods for their realization. With reference to Giddens‘ structuration theory the authors explain that the selection of methods for each individual project depends on the organization’s structure and at the same time will shape it by the actions taken during the project. Figure 13 illustrates the position of the four thematic components between the action of „IS development and deployment” on the one side and the „Organization” on the other. The list of potentially suitable methods to be used within the framework of Multiview2 includes those discussed in this section such as SSM, ETHICS, PD [Avison et al. (1998), p. 131].

The discussion in section 4.4 will point out that although organizational aspects are included in the framework of Multiview2, their role is a passive one. Thus the process model for CSCW-projects illustrated in Figure 11 exceeds the framework of Multiview2 by foreseeing tasks that aim at changing the organization and suggesting methodological support for these tasks. But since Multiview2 is an open framework which includes different types of methods and techniques, it would be possible to integrate the methods suggested for documenting socio-technical self-descriptions into the framework of Multiview2 and thereby enlarge its repertoire towards supporting organizational change.

Scenario-Based Design

> *„The defining property of a scenario is that it projects a concrete description of activity that the user engages in when performing a specific task, a description sufficiently detailed so that design implications can be inferred and reasoned about. Using scenarios in system development helps keep the future use of the envisioned system in view as the system is designed and implemented; it makes use concrete – which makes it easier to discuss use and to design use.”*

[Carroll (1995), p. 3]. Although it originates in HCI-research and not in CSCW-research, scenario based design is included here for two reasons. First, scenario-based design is not limited to the cognitive perspectives of HCI but extends well to an organizational point of view [Carroll (1997)]. Second, the idea of using design representations that include information about the technical system as well as about its context of use matches my own concerns very well.

Before summarizing the ideas behind Carroll‘s scenario-based design, a few words about the term „scenario” might be necessary in order to avoid confusion. Within software engineering the term scenario is often found in context of use cases [Kruchten (2004), p. 102]:

> *„An instance of a use-case class is also called a scenario. … Scenarios are used in the process to extract and emphasize a unique sequence of actions or a ‘thread’ through a use case.”*

Scenarios in this sense detail more generic use cases, which in turn serve the purpose of completely describing all possible interactions between a user and the software to be developed. The paradigm of scenario-based design uses a different understanding of „scenario”. Carroll‘s basic assumption is that the context of use can never be defined completely for any software. So instead of describing all possible interactions, he suggests using scenarios to illustrate the context of use by means of detailed examples which do not claim to be complete but which are able to draw the designer’s attention to central issues. Scenarios in this sense are not restricted to describe interactions between the user and the software but include non-technical activities to provide the full picture of a situation. With respect to use cases, Carroll says [(2000), p. 237]:

> *„…, one should understand use cases as simplified views of task scenarios, simplifications that are suitable for particular roles in software design. They provide the critical link between scenarios as descriptions of use and scenarios as descriptions of software to support use."*

With scenario-based design Carroll strives to elaborate a design method that does not concentrate on the technical artifact only but that takes the broader view on its use. For this purpose he suggests taking scenarios and not technical specifications or prototypes as design representations. *„The key to integrating software design with task specification is the common design language of scenarios."* [Carroll (2000), p. 253]. In this sense scenarios are artifacts that support the communication between the participants of an IT-project and that direct their attention to the use of the software. Carroll does not describe a complete process model how to organize IT-projects according to scenario-based design. But he does describe methodological components that help organizing the process of communications. The so-called *„claims"* [Carroll (2000), p. 87] e.g. serve the purpose of systematically discussing the design implications of a scenario and their pro's and con's. Lists of questions to evaluate scenarios are included as well as typologies of scenarios and their potential use within a project.

Scenario-based design relates to other approaches presented in this section e.g. by turning to PD or ethnographical methods to identify relevant scenarios.

Both, the concept of socio-technical self-descriptions and scenario based design share the idea of documenting the use of an IT-system, although they came to the idea from different sources. Carroll wanted to solve the problem of designing good interactive applications, while I strive to support groups in adopting CSCW-systems. The analysis of the case studies in section 7.1.1 provides detailed examples how use can be documented for CSCW-systems and section 7.1.2 (p. 192) discusses how this descriptions differs from scenarios in Carroll's sense.

STEPS

STEPS is a German acronym that stands for *„Softwaretechnik für Evolutionäre Partizipative Systemgestaltung"*. The English translation is „Software engineering for evolutionary participative system development". Floyd characterizes it as follows: methodological approach, which regards software engineering as an integrative part of organizational development [Floyd et al (1997), p. 13]. STEPS is relevant because it provides a process model for software engineering that includes tasks related to organizational change and this model has influenced the process model for CSCW-projects as it is illustrated in Figure 11.

Within STEPS a development process leads to a „system version" which is defined as follows [Floyd, Reisin, Schmidt (1989), p. 50]:

> *„A system version consists of technical parts in the form of executable computer programs and their defining documents, supplemented by guidelines for the organization of work to be supported by the programs."*

STEPS is based on a cyclical project model (cp. Figure 14), which foresees an early application of prototypes, in order to receive early feedback from the users that can be taken into account for the next iteration of the prototype. There are three tasks involved in the production of a system. First, „System design", which is a cooperative task performed cooperatively by software engineers and users. Two separate tasks follow: software implementation for the software engineers and „embedment preparation" for the users. The latter includes qualification as well as necessary organizational preparations.

The fact that as early as 1989 a process model for software engineering existed that included tasks for which not software engineers but users bear the responsibility, encouraged my process model for CSCW-projects shown in Figure 11 and Figure 59. The recommendations given in section 8.2 will also include a division of labor between software engineers and members of the organization.[23]

There are, however, two important aspects in which my process model for CSCW-projects transcends the STEPS model.

The first one concerns the concretion of the organizational subtask: „embedment preparation" as described for STEPS puts „organizational measures" [Floyd et al (1997), p.14] and „organizational change" in one row with qualification of users and alternations in room arrangements [Floyd, Reisin, Schmidt (1989), p. 58]. Being a generic process model, STEPS does not provide methodological support for the „organizational measures" to be taken. Especially for the CSCW-systems it seems necessary to put more emphasis on the „organizational measures". The task of *planning and establishing work procedures* is therefore explicitly included in the process model for CSCW-projects (Figure 11); and the usage of documents for socio-technical self-descriptions adds methodological support for this task (cf. chapter 5).

Second, STEPS' evolutionary approach with respect to organizational change seems to be problematic. STEPS foresees no feedback loop between the organizational task of „embedment preparation" and the technical task of „software realization". It is assumed that once the „system specification" has been agreed upon, „software realization" as well as „embedment preparation" can be performed without further feedback. Feedback that leads to a modification of the software is gathered during „use". My concepts regard the subtask of *Supporting organizational change* not merely as executing a „system specification" but rather as being interwoven with the generation of a specification and its implementation. Especially the workshop design of STWT (cf. section 5.2.3) aims at supporting feedback loops between the processes of organizational change and software implementation. Here, my concepts suggest using socio-technical self-descriptions in the form diagrams and discussing future, technically supported work procedures before they become real. For this, the process model for CSCW-projects includes a link between the organizational subtask of *planning and establishing work procedures* and the technical subtask of *System design*. Documents of socio-technical self-descriptions will relate to (intermediate) results from the process of software implementation; section 7.2.2 provides examples and discusses the experience made during the case studies.

[23] This discussion also relates to the criticism of the deployment phase as it is described for the RUP (cf. section 4.1.7)

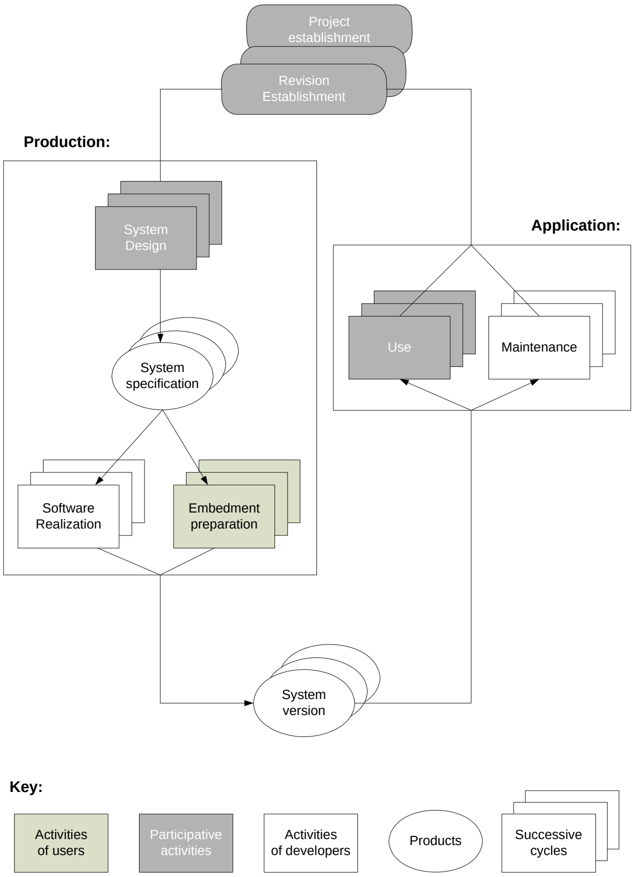


Figure 14: STEPS' Cyclical Process Model

[Floyd, Reisin, Schmidt (1989), p. 57]

4.3.4 Relation to Business Process Reengineering (BPR)

In Germany, scientists and practitioners work on standardizing the term business process and the current suggestion is to define a business-process as a sequence of logically related, productive activities that transform inputs to adequate outputs [PAS (2003), p. 41]. A similar definition is given in [Sharp, McDermott (2001), p. 43]:

> *„Businesses organize people, resources, and activities into processes which deliver value to external or internal customers, in keeping with the mission."*

BPR provides methods supporting a company's management in taking a process oriented view on the organization and redesigning complete business processes in order to make them more effective [Sharp, McDermott (2001)]. A definition of BPR that brings this intention to the point is [Hammer, Champy (1994), p. 32]:

> *"'Reengineering', properly, is 'the fundamental rethinking and radical redesign of business processes to achieve dramatic improvements in critical, contemporary measures of performance, such as cost, quality, service, and speed.'"*

There is a complicated relation between BPR and CSCW-projects. On the one hand BPR belongs to a different discipline, namely economics, which has different goals and provides different methods than computer sciences in general and CSCW in particular.

Integrating IT-systems into the business processes of organizations is not the primary goal of BPR, but there are aspects that closely relate BPR and CSCW:

1) BPR is often related to IT-projects for two reasons:
   a) IT is regarded as an „enabler" of business processes ([Sharp, McDermott (2001), p. 333], [Hammer, Champy (1994), pp. 83]).
   b) Business process modeling can be used to support requirements elicitation (cf. discussion of RUP in section 4.1.6).
2) Both regard processes: CSCW regards work procedures, BPR business processes.
3) Both employ graphical representations of these processes: within CSCW the methods range from rich pictures (cf. section 4.3.3, p. 118) to business process descriptions in workflow projects; a prominent example for BPR is the usage of event-process-chains [Scheer (1991)].

The question for this thesis is whether business process models are socio-technical self-descriptions and whether BPR is a means for supporting structural coupling within a CSCW-project. Consider the following statement [Hammer, Champy (1994), pp. 85]:

> *"The fundamental error that most companies commit when they look at technology is to view it through the lens of their existing processes. They ask, 'How can we use these new technological capabilities to enhance or streamline or improve what we are already doing?' Instead, they should be asking, 'How can we use technology to allow us to do things that we are not already doing?'"*

The approach that new processes that include new technology need to be invented and elaborated is in line with my own approach for supporting CSCW-projects. With regard to Figure 6 and the discussion of self-descriptive artifacts in sections 2.4.1 and 3.1.3 business process models can well be part of an organization's self-description as long as they result from communications within the organization. The analysis of the case studies (section 7.1.2) includes a discussion how the documents for socio-technical self-description used in CSCW-projects differ from business process models created in one of the case studies. It will turn out that it is a question of focus and abstraction regarding the documentation and a question of method regarding the creation and usage of the documentation.

The modification of complete business processes is certainly one aspect of structural coupling, therefore methods of BPR could be employed in the subtask *Supporting organizational change / structural coupling* (cf. Figure 11).

However, BPR is often connected with management approaches that do not share the theoretical basis of this thesis. As the term „process reengineering" implies, business processes are viewed as objects that are designed by business analysts who are either managers within a company, or even more often, external business consultants [Sharp, McDermott (2003), p. 31]. Within the RUP Kruchten created the role of business designer for this purpose (cf. section 4.1.6). So before integrating my methodological concepts with methods of BPR, one would have to check whether the particular approach to BPR is compatible with the interpretation of organizations as autopoietic systems.

Within the field of CSCW there are debates about the methods of BPR, in particular about the way graphical process models are used. The next section discusses the discourse on formal representations of work procedures within the field of CSCW.

4.3.5 Representations of Work-Procedures

Chances and Risks

Potential Benefits of Formal Representations of Work-Procedures	
	„Things are made visible so that they can be seen, talked about, and potentially, manipulated. It is the last that constitutes the power, for better and worse, of the construction of representations of work.” [Suchman (1995), p. 63]
	Suchman qualifies her criticism of process-descriptions by saying: „As long as such representations remain embedded in the doing of the work, they serve as a useful tool for organization members in their practical reasoning about and action within the organization.” [Suchman (1995), p. 61].
	Schmidt emphasizes that formal constructs can take the form of „coordinative protocols” and that „such protocols convey a precomputation of tasks interdependencies which assists actors in reducing the complexity of coordinating their activities.” [Schmidt (1999), p. 326]
	Under the headline „Process Descriptions as Organising Resources” [Dourish (2001a), p. 58] Paul Dourish refers to Suchman’s distinction between maps and scripts and continues with reference to maps: „This could certainly constitute the use of a process description as an organizing resource, and particularly, a resource for the coordination of collaborative activity.”
Potential Dangers of Formal Representations of Work-Procedures	
	„Not only do representations of work involve perspectives and interests, but work has a tendency to disappear at a distance, such that the further removed we are from the work of others, the more simplified, often stereotyped, our view of their work becomes.” [Suchman (1995), p. 59]
	Mike Robinson and Liam Bannon „develop an analytic case against an objective reality that can be usefully „captured” in a model and subsequently used as a sufficient basis on which to develop a computerized system.” [Robinson, Bannon (1991), p. 220]. They criticize „that the current generation of models over-emphasize determinism at the expense of interpretation in the work process” [ibid. p. 219].
	„Problems arise, however, when normative representations are either generated at a distance from the sites at which the work they represent goes on or taken away from those sites and used in place of working knowledge.” [Suchman (1995), p. 61]
	Suchman also points out a political dimension that becomes relevant when process descriptions determine the way people interact with each other: „The inscription of formal representations of action in technical systems transforms the debate more clearly into a contest over how our relations to each other are ordered and by whom.” [Suchman (1994), p. 188]. She presumes an „abiding interest that those committed to the reproduction of an established institutional order might have in replacing the contested moral grounds of organizational commitment and accountability with a scheme of standardized, universalistic categories, administered through technologies implemented on the desktop” [ibid].

Table 10: Potential Benefits and Dangers of Formal Representations of Work-Procedures

The use of formal representations of work-procedures – especially in the way they are used in business reengineering projects – gave rise to deep discussions within the CSCW community. The debate began with the deployment of workflow systems for office automation [Suchman (1987)] because at the heart of each workflow-engine there is a process-description that determines the behavior of the software and requests corresponding actions from the users. By and by the qualities, potential benefits and potential dangers of formal representations of work-procedures in general became subject to discussion [Herrmann et al. (2002a)]. Table 10 summarizes some of the most prominent statements and positions.

This debate is highly relevant for this thesis, because section 5.2.2 will introduce semi-structured diagrams as a suitable form for socio-technical self-descriptions. The following section introduces semi-structured diagrams and positions their use with respect to the arguments of Table 10.

Semi-structured Models

Starting in the 1980's research has been done to augment technical modeling languages such as Petri-Nets with concepts suitable for representing a user's perspective ([Oberquelle (1987)], [Luczak, Wimmer, Schuhmann (1996)]). My own workgroup has worked for years on concepts that promote the positive aspects of representations of work-procedures while diminishing their negative aspects. In the center of all approaches there is the semi-structured modeling method named SeeMe. The driving force behind SeeMe is the goal to include the contingencies of human action and communication in a modeling method ([Herrmann (1997)], [Herrmann et al (2004a)]). This goal encompasses the modeling language itself and its employment in projects. Derived from BPM-techniques such as ARIS [Scheer (1991)], SeeMe is a graphical notation including all syntactical elements necessary to describe business processes and work-flows. In addition SeeMe offers syntactical elements to express incompleteness [Herrmann, Loser (1999)]. These elements are an answer to the problem that formal representations of work-processes are not good at documenting „situated action".

The modeling method SeeMe is explained in detail in the literature cited above and a brief overview of the most important notational elements is provided in appendix 11.2 (p. 305). But since SeeMe is used in both case studies and SeeMe diagrams will be important for the discussion of the case studies in section 7, an example shall illustrate how the contingency of human action can be illustrated in form of a SeeMe diagram.

The example in Figure 15 includes a *Clerk* (roles are represented as red ellipses), the activity of *double-check the accuracy* in a *Customer form* (activities are represented as yellow rectangles with rounded edges; entities are blue rectangles) and *additional information* that are used for the task. Figure 15 includes three variants: the first one to the right is straight forward and means that the *Clerk* performs the task of *double-checking* and that he uses the *customer's form* as well as *additional information*. The second variant in the middle includes a modifier to the activity (modifiers are represented as green hexagons) stating that the task *double-check the accuracy* is performed if and only if the *Customer* is *unknown*. The third variant on the right side uses empty modifiers expressing vagueness to represent situated actions. Options are included for the *Clerk* in two places. First, it is left to his discretion whether he performs the task of *double-checking the accuracy* at all. Second, whether or not he includes *Additional Information* is also his choice.

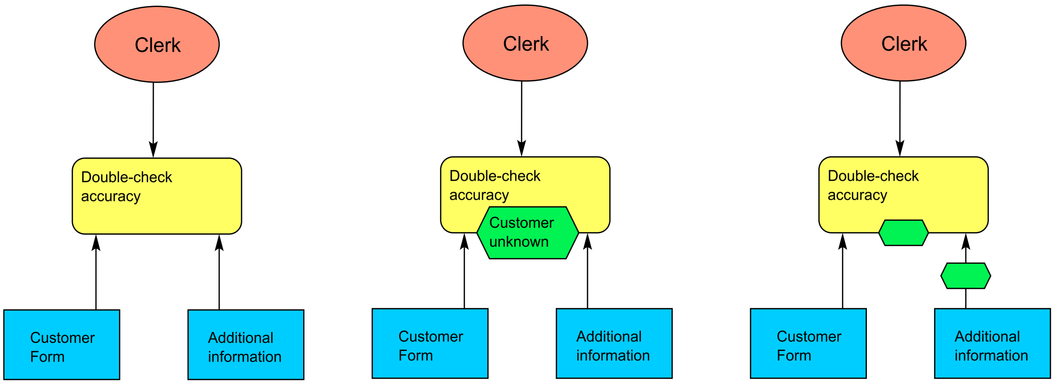


Figure 15: Representation of Situated Action with SeeMe

The potentials of the modeling language SeeMe to combine deterministic and contingent situations of working life, lead to further theoretical considerations, about the role of models within cooperative processes [Herrmann et al. (2002a)]. The insight relevant in this context is that graphical representations are suitable artifacts for supporting the planning of cooperative work-processes as long as they are integrated into an appropriate communicative context. Within the workgroup numerous case studies have been conducted to evaluate and further improve the employment of SeeMe (e.g. [Herrmann et al. (2000)], [Loser, Herrmann (2001)], [Loser, Herrmann (2002)]; for an overview: [Herrmann et al. (2004a)]). It is the conclusion from the case studies that semi-structured diagrams can successfully be used to support group processes in the context of CSCW-system-projects. Both of my case studies continued the research on using semi-structured diagrams.

4.4 Summative Discussion

Software Engineering and Systemic Intervention

In this chapter, three fields were analyzed for the support they offer with regard to the integration of organizational change and software engineering. First (section 4.1) methods and process models stemming from technical discipline of software engineering were discussed. Table 11 summarizes this discussion by listing those aspects of software engineering that provide links to the social processes of organizational change; and those aspects of software engineering that contain problematic assumptions with respect to the social qualities of organizational change.

What are the links that the discipline of software engineering offers with respect to organizational change?

1. Evolutionary approaches to software engineering allow the integration with the process of structural coupling in the organization
    a. by providing intermediate artifacts,
    b. by being able to react to changes.
2. The phase of requirements elicitation is well suited to enter processes of organizational change, because it foresees an involvement of users and a close cooperation with software designers.
3. Various software engineering methods such as the RUP rely on the usage of visual models. Such documents could serve as artifacts for communication processes within the organization.
4. Use-case-driven approach: the idea that there need to be artifacts that serve as a guiding artifacts through the whole development.
5. The inclusion of models such as business models for describing the context of use for the software.
6. The definition of a deployment discipline (e.g. in the RUP) that describes tasks necessary to transfer software from the site of development into usage.
7. The definition of operational procedures as a part of system design.

What are assumptions within software engineering that are problematic with respect to the social qualities of organizational change?

1. Assumption that the usage of visual models leads to unambiguous communication between people.
2. Assumption that diagrams can „unambiguously specify behavior"
3. Assumption that the same models that are appropriate for describing the software are also appropriate for describing the context of use.
4. Assumption that software engineers are responsible for embedding the software into a social system and that they have the means to do so.

Table 11: SWE and organizational change, Summary

Second, systemic intervention was introduced and discussed (section 4.2). Systemic intervention provides methods for supporting organizational change which are founded on system theory as it is introduced in chapter 2. Here as well a table (Table 12) summarizes the potential links and problems regarding integration with software engineering.

Third, methods from the field of CSCW, that explicitly aim at supporting organizational and technical issues, were analyzed (cf. section 4.3). The findings with regard to these methods are summarized in the following two subsections.

What are the links that the discipline of systemic intervention offers with respect to software engineering?
1. Documented self-descriptions of organizations that are used to stabilize the system's behavior and make it foreseeable can be used within the SWE-process (e.g. for requirements elicitation). 2. CSCW-systems influence the way people communicate with one another; systemic intervention provides methods to identify and change patterns of communication. 3. From a systemic point of view, the members of an organization are responsible for bringing about organizational change; this task cannot be delegated to external consultants. This concept links well to the idea of actively involving users in PD (participatory design). 4. The interdependence between individual and organizational learning links to the fact that training is already an integral part of software engineering projects.
What are assumptions within systemic intervention that are problematic with respect to its integration with software engineering within a CSCW-project?
1. Systemic intervention understood as "observing observations" does not directly relate to issues of using a CSCW-system. 2. Diagnosis / analysis of status quo are viewed as interventions with the goal to serve the organization not the consultant.

Table 12: Systemic Intervention and SWE, Summary

Support for the Design of a CSCW-Application

Looking at the kind of support the methods discussed in the previous section provide for CSCW-projects, one can conclude, that a plentitude of methods and techniques is available for the design of a CSCW-application. These might be categorized as follows:

1. Some methods such as ETHICS include psychological instruments to identify organizational issues to be taken into account when designing a CSCW-system.
2. Practically all methods foresee the direct participation of the future users in the design process to make sure that their knowledge is included in the design.
3. Some methods emphasize that requirements for a CSCW-system are constructed socially rather than existing objectively. And that therefore the design process is a communication process between technical experts, stakeholders in general and future users in particular. Approaches such as PD, JAD, CD, SSM, and scenario-based design provide methods and artifacts to support this communicative process.
4. Methods such as SSM, JAD but also ETHICS regard information system development as only one part within a larger project that aims to solve an organizational problem. They provide techniques to pass information from the larger project into the technical development process.
5. Methods that suggest special design representations such that the designers do not forget about the context of use. Scenarios in the sense of scenario-based design are the best example; but rich pictures as described in SSM also contribute to this category.

With regard to the process model in Figure 11 these methods are strong in supporting *system design*.

Support for Structural Coupling

But, what about the task of *supporting organizational change / structural coupling,* which is also included in the process model for CSCW-projects in Figure 11? The support provided by the methods discussed is not as abundant.

There are those methods that are not meant to support processes of organizational change at all. CD is one example. Since it is intended for designers of standard software, it does not include methods for supporting change processes within an individual organization. Multiview2 is difficult to evaluate: its basic concept as it is illustrated in Figure 13 claims to influence the organization's structure. However, of its four components, only „software development" is constructive, the others (including socio-technical and organizational) are for analytical purposes only. This would imply that organizational change is supported only indirectly by the way of designing an IT-system. What it missing in the framework are methods that actively support the structural coupling.

Scenario-based design is a method with the potential of supporting organizational change, because its design representations include the context of use. But due to its roots in HCI it does not systematically support the processes necessary to organize computer supported cooperative work, although Carroll [(1997)] argues convincingly that scenario-based design is not limited to the cognitive perspective. It is not obvious that scenarios are a suitable means to support the agreements necessary for cooperative work (cf. discussion pp. 120).

Methods such as SSM, ETHICS and Eason's framework provide techniques for actively supporting organizational change. What is missing however is the support for structural coupling, which is the support for organizational change in response to features of a CSCW-system. SSM does not address the issue of modifying purposeful action systems in response to a new CSCW-system. It does describe how the definition of a purposeful action system helps in designing an appropriate IT-system, but it falls short in supporting processes of interdependent change. Also, SSM does not relate its methodological components systematically to software engineering processes. It is only mentioned that the usage of prototypes within an iterative approach to software engineering is compatible to the ideas of SSM [Checkland, Holwell (1998), pp. 114].[24] ETHICS incorporates the original socio-technical paradigm of „joint optimization" stating that organizational and technical systems are interdependent and must therefore be „designed" in an integrative effort. However, Mumford only describes that the results achieved by her instruments are used to inform the selection of a „strategy" and then the „evaluation" of the operational system (cf. Table 5, p.111). She remains quite vague about how the steps „designing the system in detail" and „implementing the system" are intertwined with the evolvement of work-procedures. The tools provided by ETHICS (such as the variance analysis tool or the questionnaire) do not relate to the use of CSCW-systems [Mumford (1995), p. 53]. MUST addresses the planning of new computer supported work-procedures but it is strong only in the early (design-) phases

[24] SISTeM is a method "integrating informational and organizational development" [Atkinson (2000a), p. 104] that is based on Checkland's SSM and includes links to artifacts such as dataflow diagrams or use case diagrams. However, the author then switches from system theory to actor network theory to explain human action in a network of human and non-human actants. SISTeM is then described on a level of abstraction that forbids a comparison with the other methods in this section.

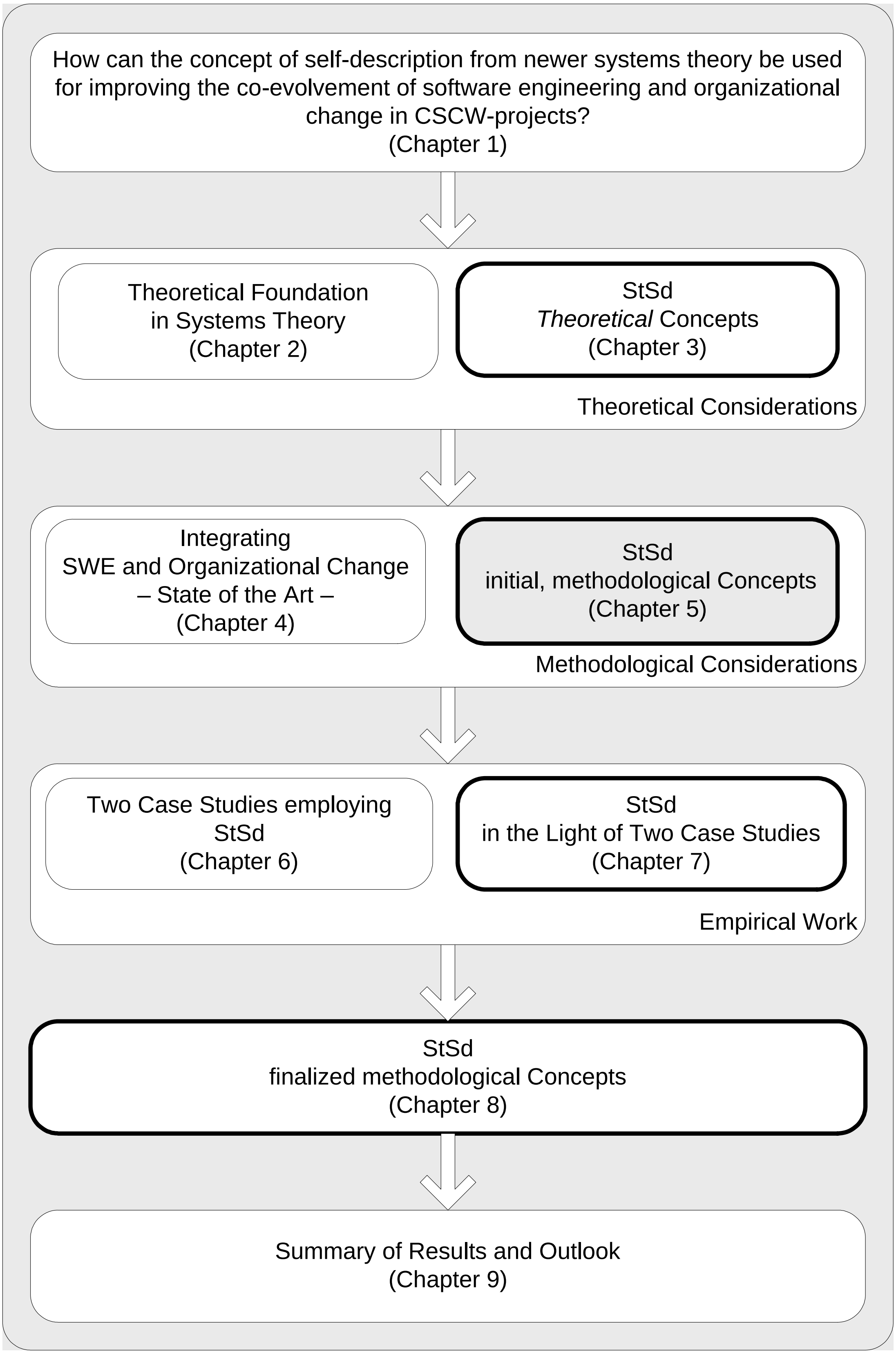
of an IT-project. It lacks methodological support for the implementation and deployment phases of the software. MUST foresees actions to develop and anchor *„visions of the overall change"*, but it does not support the actual process of change. The workshop designs suggested in PD as well as JAD support the integrated discussion of technical and organizational issues. However they need to be incorporated into a larger process model organizing CSCW-projects; and they could benefit from artifacts explicitly integrating descriptions of the CSCW-application and descriptions of work procedures. STEPS provides a process model for software engineering that does foresee processes of organizational change systematically; it does relate them to the software being developed; and it does encompass more than the design phase. It is the strength of STEPS to systematically include issues of organizational change into a process model for software engineering. My process model transcends STEPS by detailing the social process of *planning and establishing work procedures* and providing feedback links to the *technical design and implementation of software* (cf. section 4.3.3, p. 121).

Conclusions

The described methods contain numerous insights that influence my initial concept for supporting socio-technical self-descriptions; these have been discussed in detail in the sections 4.1, 4.2, and 4.3. But there are three main ideas that could be called the essence of my review of existing methods on which the initial concept is based:

- A CSCW-project should have a guiding artifact that describes the context of use rather than focusing on the CSCW-system alone: Therefore, documented socio-technical self-descriptions will be used as guiding artifacts that integrate technical components with their context of use (cf. section 5.1).
- The design of a CSCW-system and corresponding work procedures is a social and hence a communicative process, in which the perspectives of many stakeholders need to be integrated: Therefore, I elaborate the workshop design STWT that supports communication about the CSCW-system in its context of use (section 5.2.3).
- The technical design and the process of organizational change are interdependent: Therefore, the support for socio-technical self-description is integrated into a process model for CSCW-projects which foresees the "joint optimization" of organization and CSCW-system (Figure 11 and Figure 59).

The methodological support for socio-technical self-descriptions transcends the discussed methods by explicitly supporting the process of structural coupling in parallel with software design and implementation.


How can the concept of self-description from newer systems theory be used for improving the co-evolvement of software engineering and organizational change in CSCW-projects?
(Chapter 1)
Theoretical Foundation in Systems Theory
(Chapter 2)
StSd
Theoretical Concepts
(Chapter 3)
Theoretical Considerations
Integrating SWE and Organizational Change – State of the Art –
(Chapter 4)
StSd
initial, methodological Concepts
(Chapter 5)
Methodological Considerations
Two Case Studies employing StSd
(Chapter 6)
StSd
in the Light of Two Case Studies
(Chapter 7)
Empirical Work
StSd
finalized methodological Concepts
(Chapter 8)
Summary of Results and Outlook
(Chapter 9)

5 StSd – Initial Methodological Concepts

Based on the theoretical concepts I elaborated in chapter 3 as well as the best practices and methodological gaps I identified in chapter 4, this chapter derives my initial methodological concepts for supporting socio-technical self-descriptions in CSCW-projects. These initial concepts are used in the two explorative case studies (chapters 7, 8) and will lead to a detailed set of recommendations documented in chapter 8.

5.1 Basic Concept

5.1.1 Vision and Path

What do I want to achieve with the methodological concept that is described in this chapter and applied in the two case studies? I want to avoid that the CSCW-systems developed in the case studies cause problems of the kind cited at the very beginning in section 1.1.1.

Or, to put it positively: My goal is that each case study should result in a socio-technical setting in the sense of Term 22. There should be a fully functional CSCW-system matched by work procedures within the organization that make expedient use of this system. The members of the organization should know about the system and their ideas about it should be congruent such that a coordinated use of the CSCW-system is possible.

To reach this goal, I transfer the theoretical concepts of chapter 3 into methodological components to be applied in the case studies. The central ideas are:

a) Regarding the organization as an operationally closed system: the use of a CSCW-system is neither determined by the CSCW-system itself nor can it be prescribed by a design team. Therefore a process is included into the CSCW-projects during which the members of the organization plan their own computer supported cooperative work.

b) The initiation of organizational change early in the project: members of the organization are supported in elaborating their new, computer supported work procedures, while the CSCW-system is being developed.

c) Creation and maintenance of socio-technical self-description: the organization is supported in including the usage of the CSCW-system into its self-descriptions and in documenting the usage of the CSCW-system in form of socio-technical self-descriptions.

The support for creating and maintaining socio-technical self-descriptions is at the center of my methodological concept. The main arguments from sections 2.4 and 4.2 shall be briefly repeated:

- Every organization needs self-descriptions for maintaining its identity.
- Organizations that conduct explicit self-descriptive communications are more likely to develop complex structures than those that do not;
- Self-descriptions serve the dual purpose of stabilizing an organization by defining its identity and enabling change by offering an object for reflection.

Figure 6 and the related discussion in chapter 3 provide an overview of the various manifestations of (socio-technical) self-descriptions in organizations. For the purposes of developing a method for CSCW-projects, I will concentrate on three forms of StSd which are discussed in the following section.

5.1.2 Forms of StSd in CSCW-Projects

Figure 16 illustrates three levels of self-descriptions that will be relevant for the methodological support.

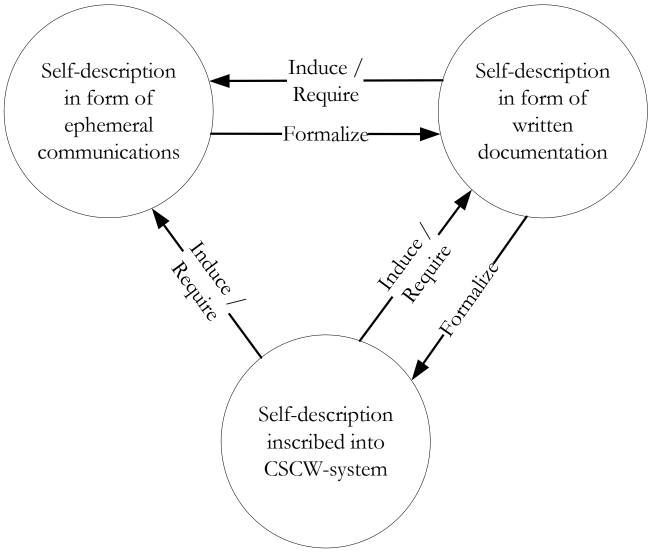

Self-description in form of ephemeral communications
Self-description in form of written documentation
Self-description inscribed into CSCW-system
Induce / Require
Formalize
Induce / Require
Induce / Require
Formalize

Figure 16: Three Forms of Socio-Technical Self-Descriptions

In any organization there are two levels of self-descriptions: ephemeral ones and those that are made durable by writing. In general, this distinction coincides with the distinction between informal and formal. The official and formal aspects of the self-description such as the purpose of the organization, its values and its hierarchy are preserved in documents. Additionally there are agreements about how the daily work is done and how departments cooperate that are at least as important for the organization's functioning but that do not exist in writing. In Figure 16 these two types of self-descriptions are labeled *„Self-description in form of ephemeral communication"* and *„Self-description in form of written documentation"*. There are exchange processes between the two levels. Informal agreements may become so important that at some point they are written down and thereby become formal rules. And every (new) formal rule needs additional informal agreements to fit into the communicative context and to allow the handling of exceptions. In socio-technical organizations there is one additional level of self-descriptions: those that are inscribed into the CSCW-system (cf. section 3.1.3, p. 73).

When working with StSd not only all three levels but also the exchange processes in-between these levels have to be taken into account: Rules can find their way from informal agreements via written statements into software-coded functions. And the other way around: Software functions need additional rules to ensure their coordinated use in a working context. These additional rules can be developed and documented in parallel with the software development or they can evolve slowly during usage as informal agreements that may or may not be documented. In an ideal socio-technical system all three levels of self-description are consistent within themselves. The formal rules as well as the rules that became part of the software functions resulted from communications within the socio-technical system. Informal agreements are added as appropriate.

Distributed Nature as Inevitable Characteristic of StSd

It is important to note that the distributed nature of StSd is an inevitable characteristic and not a flaw to be corrected. One might argue that all rules necessary to describe a work procedure can be coded into software as long as they are properly described as requirements. Then no further documentation or agreements describing the integration of the software into an organizational context would be needed. There are two arguments against this position. First it is simply not possible to completely describe complex cooperative working processes and inscribe them into a CSCW-system. There will always be exceptions that make additional agreements necessary. Second, work-procedures inscribed into software are rigid because events and conditions are firmly linked to further activities. It is a general rule that applies to the social as well as to the technical world that fixed couplings always need to be embedded into loose couplings. In mechanics this is done by installing flexible links to avoid breaking; in real-time software-systems data-buffers are implemented to decouple processes; and in organizational procedures informal agreements are needed to apply formal rules (cp. [Luhmann (2000), p. 375]).

Focus of this Thesis

With regard to CSCW-projects this implies that all three levels of StSd will be encountered: there will be the workflows encoded into the software. There will be informal agreements incorporating the CSCW-system into work procedures and there will be documents describing the usage of the CSCW-system on a more formal level.

The last aspect is my focus in this thesis. By creating written self-descriptions about the usage of a CSCW-system, its integration into work-procedures can be subject to deliberate planning. It is discussed in chapter 4 how existing methods fall short in supporting the structural coupling during a CSCW-project. The following sections present the initial methodological concepts for supporting socio-technical self-descriptions in CSCW-projects with the goal to initiate the process of structural coupling between organization and CSCW-system.

5.2 Socio-Technical Self-Descriptions in CSCW-Projects

For each of the following topics it is discussed how it manifests itself within a CSCW-project:

a) The **content** of socio-technical self-descriptions (section 5.2.1)

b) The **form** of their documentation (section 5.2.2)

c) **Workshop design** for creating and maintaining socio-technical self-descriptions (section 5.2.3)

d) Socio-technical self-descriptions as a means for **integrating** the processes of software engineering and organizational change (section 5.2.4)

In preparation of the analysis of the case studies research questions for all four topics are derived; they are summarized in Table 14 as a reference for reading chapter 7.

5.2.1 StSd – Content

Initial Concept

Self-descriptions as they are defined in Term 11 describe the actions that are expected from the members of an organization. Socio-technical self-descriptions add the relation between these actions and a CSCW-system (cf. Term 21). But how exactly will the actions and the usage of the CSCW-system be described?

The initial concept is very open in this respect; the goal is to document anything that seems to be necessary for supporting the organization in describing its future computer supported work procedures. The resulting documents will then be analyzed for deriving recommendations with regard to the content of StSd. Nevertheless the theoretical considerations as well as best practices from existing methods in the field of CSCW provide some initial ideas regarding the content of StSd:

- StSd should describe the context of use rather than focus on the CSCW-system (cf. scenario based design, pp. 120).
- StSd should document the division of labor / interaction between automated processes and human action (cf. Eason’s socio-technical design, pp. 112).
- StSd should describe the organization as an activity system (cf. Luhmann‘s system theory, section 2.3.4).

- StSd should describe the CSCW-system from a users' perspective, because they need to be interpreted by members of the organization.

Content-Related Research Questions for the Case Studies

It is a major goal for the analysis of the case studies to provide a more detailed description of the contents of StSd. So the first content-related research question asks: What topics were discussed and documented that contribute to the self-description of the organization with respect to the new technical system (cp. Research question C1 in Table 14)?

From a skeptical point of view one could say that no socio-technical documentation needs to be added to use-cases (cf. section 4.1.5), business-models (cf. section 4.3.4) or scenarios (cf. section 4.3.3, p. 120), because they contain everything that is noteworthy. Therefore it is another goal of the content-analysis to justify the approach of using StSd by discussing how these topics exceed the scope of the other forms of documentation (cp. Research question C2 in Table 14).

The method of action research brings about the involvement of the researcher in the intervention. This proceeding bears the danger that I myself give rise to certain topics which are of special interest to me. If this were the case, the list of topics gathered could not be used to justify a new type of documentation as intended. Therefore the analysis of the case-studies will have to show how topics were found during the workshops (cp. Research question C3 in Table 14).

For the same reason it is interesting to ask whether the participants considered it useful to have additional information about the integration of a new CSCW-system into their work-procedures available. The general task of producing documentation describing the socio-technical system was induced by me, so it is not self-understood that the participants also attached any importance to them (cp. Research question C4 in Table 14).

5.2.2 StSd – Form of Documentation

Options for the Form of StSd

The next step is to decide on a form in which socio-technical self-descriptions should be documented. There are several ways to argue for the need of a written representation of socio-technical self-descriptions. The first is to use system theory and cite Luhmann: self-descriptions may not be limited to ephemeral communications but need to be captured in an enduring format, because otherwise a modern organization could not use it to maintain its identity (cf. section 2.4, p. 50). A second line of argumentation is that socio-technical self-descriptions are meant to support the integration of organizational change and software engineering by serving as a boundary object. Section 3.1.3 (p. 72) explains the special qualities of boundary objects that are stable enough to maintain an identity and at the same time flexible enough to be used by different groups of actors. Written documents are one way to provide the needed stability.

Numerous forms of representational artifacts have been discussed in this thesis so far. Which ones are appropriate for socio-technical self-descriptions? To decide this question, it is helpful to describe the differences between the suggested forms. Figure 17 provides a categorization on two dimensions:

a) Verbal vs. Graphic

b) Formal vs. informal

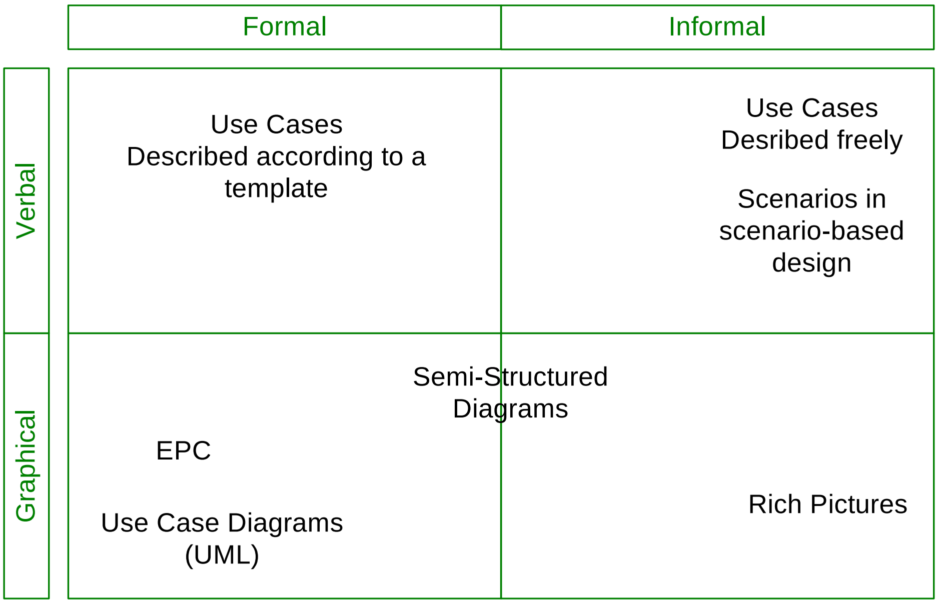


Figure 17: Possible Forms of Artifacts for StSd

Software engineering knows verbal as well as graphical representations. UML provides a set of graphical representations (cf. section 4.1.4); use case diagrams and activity diagrams are examples included in Figure 17. All diagrams described by the UML are considered as formal because they have a defined syntax. Use cases themselves are usually described using plain text (cf. section 4.1.5); therefore they are located in the upper part of Figure 17. But this does not mean that use cases cannot be formal: Some authors suggest templates for the description of use cases [Cockburn (2001)]. Both types of use case-description are included in Figure 17. EPCs are examples of formal diagrams that are used for business-process reengineering (cf. section 4.3.4). Rich pictures which were introduced in connection with SSM (cf. section4.3.3, p. 118) are an example for informal graphics (i.e. diagrams without syntax). As a response to the criticism of formal representations of work (cf. section 4.3.5) semi-structured diagrams were introduced to combine the formal aspects, needed for the representation of predefined procedures or technical systems, with vague aspects characterizing human activity (cf. section 4.3.5, p. 126).

In principle, all of the artifacts included in Figure 17 could be used for documenting socio-technical self-descriptions. The choice for one is guided by two requirements. First, socio-technical self-descriptions need to be embedded into a communicative process, which will be provided in form of a workshop design (cf. section 5.2.3). Diagrams are considered superior to text because they are better suited for supporting group discussions. Second, socio-technical self-descriptions need to integrate the representation of contingent human action and technical components; for this, semi-structured diagrams have proven their worth (cf. section 4.3.5).

From the diagrams listed in Figure 17 semi-structured diagrams in the modeling language SeeMe are chosen. As discussed in section 4.3.5 there is an abundant source of positive experience within my workgroup in using SeeMe in the context of IT-projects. It has been thoroughly examined and shown that SeeMe diagrams can be understood and created by people who have no background in computer science [Loser (2005), pp. 143]. And as a result of previous research an editor is available that supports the discussion and modification of diagrams within a workshop setting. This editor (called SeeMe-Editor) has been designed in line with requirements derived from previous case studies [Loser (2005), pp. 201], but it has never been employed within projects before. The case studies were used for continuing the research on semi-structured diagrams in combination with evaluating the SeeMe-editor.

By this decision for using SeeMe it is not ruled out that other modeling languages could also be used for documenting socio-technical self-descriptions. But when choosing an appropriate modeling language one has to keep in mind that the diagrams need to serve as a boundary object between processes of organizational change and software engineering. Informal sketches such as rich pictures bear the danger of being not rigid enough to transport rules for incorporating the CSCW-system into work procedures. Formal notations such as Petri-net based EPCs on the other hand may be too restricting and bear the danger of simplifying contingent situations.

Introductions into the SeeMe modeling language can be found in [Loser (2005), chapter 4.4, pp. 85], [Herrmann et al (2004a)], [Herrmann, Loser (1999)]; and appendix 11.2 provides a summary of the basic notational elements. Further explanations are provided with the discussions of the case studies in chapter 7. But two aspects which are of special relevance for this thesis shall be discussed in the following sections.

SeeMe for representing CSCW

How is a technically supported, work-related activity represented in SeeMe? Figure 18 shows various examples. The construct labeled a) to the left shows the basic form: an activity named *workrelated activity* is symbolized by a yellow rectangle with rounded edges and is connected to an entity named *CSCW-System* represented by a blue rectangle. The arrow goes from the entity to the activity representing the relation „is used by".

The construct labeled b) to the right in Figure 18 shows *Workrelated Activity* consisting of three sub-activities and the *CSCW-System* consisting of three subsystems. The relations between these sub-elements vary: *Activity 1* is supported by *Subsystem 1*. *Subsystem 2* does support the overall activity *Workrelated Activity*, however it is left open whether all, some or none of the sub-activities make use of *Subsystem 2*. This is a typical example for a supporting function such as a help-system. It provides information applicable to *Workrelated Activity*, but whether it is indeed used or not depends on the specific context. *Activity 3* could make use of *CSCW-System* but it is left open, whether all, some or none of the subsystems are used. An example for this is the organization of a small internal meeting. This activity could make use of CSCW-functions such as group calendar and messaging; but some meetings are so small or at such a short notice, that the responsible person decides to talk personally to the participants.

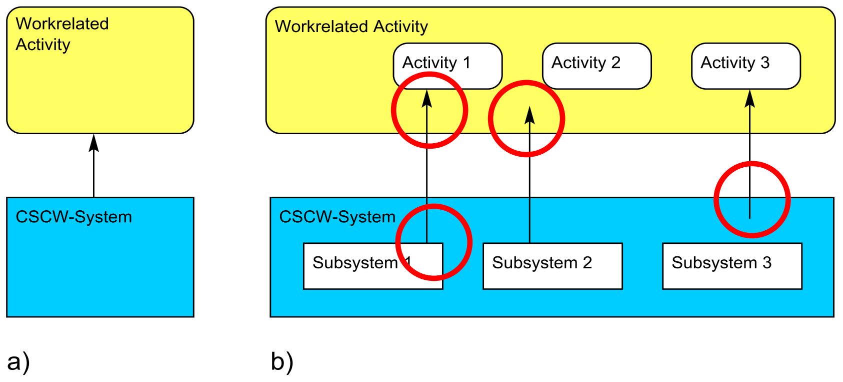


Figure 18: SeeMe Constructs Showing Human Activity Supported by a CSCW-System

SeeMe distinguishes between Human Activity and Automated Processes

If a description of computer supported cooperative work is intended to support the organization of such work, then it is essential to differentiate between activities performed by human actors and those that performed automatically by the CSCW-system[25]. The two different SeeMe-constructs are shown in Figure 19. Part a) shows a role, symbolized by a red ellipse, connected to an activity. The arrow starting at the role and ending at the activity represents the relation „does". It is important to note that within SeeMe „roles" are defined in a sociological sense. A role can be any particular person or a group of persons; a role can also refer to an abstract construct such as a department or a company as a whole; however a role never refers to technical agents. The latter will always be represented by entities.

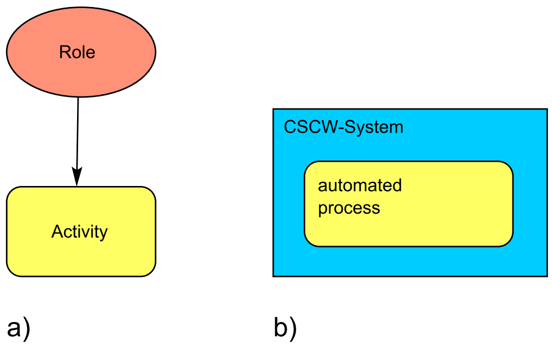


Figure 19: SeeMe Constructs Representing Human Activity and Automated Processes

[25] If representations are intended to analyze and explain certain phenomena in socio-technical settings, it can be worthwhile not to include this distinction; actor-networks are one example (cf. section 1.3.4, p.17).

Part b) in Figure 19 shows such an entity *CSCW-System.* To represent activities performed by a technical agent such as a CSCW-system, symbols for activities are embedded into the entity representing the corresponding technical system. An example for such an automated function is a consistency check after data has been entered by the user.

Form-Related Research Question for the Case Studies

The analysis of the case studies has to discuss whether the choice of semi-structured diagrams was a wise one. Loser has demonstrated that users with no training as software engineers or programmers are able to interpret semi-structured diagrams [Loser (2005), chapter 7], and that these diagrams are helpful in the context of participative design-processes. Loser's research is directed toward the adoption of commercial-off-the-shelf (COTS) software. My research questions focus on the problem whether semi-structured diagrams are suitable for holding parts of the socio-technical self-description in the context of software-development. The experiences made with documents in the form of semi-structured diagrams have to be analyzed. Were they suitable to capture the topics that came up during the project (Research question F1 in Table 14)?

Another research question refers to the representation of the CSCW-system within the socio-technical diagrams. Case studies affirm that semi-structured diagrams are useful to document contingent, technically supported work-procedures (cp. Section 4.3.5); however the representation of the IT-system has not yet been subject to systematic evaluation. Therefore another research question is how the CSCW-system can be represented in the semi-structured diagrams (Research Question F2 in Table 14). Since the CSCW-systems are being developed within the projects, it is also interesting to see whether and how this representation changes over time.

5.2.3 Workshop Design STWT

Workshops as Communicative Settings

The methodological support for socio-technical self-descriptions includes a workshop design named socio-technical walkthrough (STWT). Before discussing the particular characteristics of the STWT, the reasoning behind the decision to include workshops in the methodological repertoire shall be recapitulated:

- The analysis of existing methods for CSCW-projects in section 4.4 identified the support for communication processes as an important element of methodological support. Facilitated workshops are a proven means for providing this support and will therefore be used within the framework for supporting socio-technical self-descriptions.
- One result of the theoretical discussions in chapters 2 and 3 was that self-descriptive artifacts need to be embedded in a communicative context, because otherwise they would loose their self-descriptive characteristic for the organization. The workshop design STWT provides such a communicative context.

- It was said in section 2.4.3 that self-descriptions should be set apart from „normal" workrelated activities. Workshops support this insight by providing a setting in which people leave their normal work environment to discuss some issue; in the case of STWT during a CSCW-project the issue is the organization of their computer supported work.
- It was said in section 2.4.2 that self-descriptions should also exist in form of durable documentation. Workshops in general are settings suitable for elaborating such results. The STWT provides specific techniques for creating, discussing and changing artifacts of socio-technical self-description.

The fore standing arguments support the decision for using workshops; but they do not yet substantiate the need for a specific workshop design. Workshop designs are templates or storyboards which facilitators may use for structuring a workshop; the STWT is a specific workshop design for creating and discussing socio-technical self-descriptions in the context of CSCW-projects.

STWT – Workshop Design for Socio-technical Self-descriptions

Prototyping has become a standard means within SWE: it provides artifacts to which users can relate and which can be used for discussing and designing features of the CSCW-system. The work-procedures corresponding to the technical CSCW-system do not lend themselves to prototyping easily. Scenarios or role-games only catch a small section of reality; and one cannot simply ask an organization to try certain procedures for one week and then test other procedures for another week. The STWT addresses this problem: Within the STWT it is foreseen to represent technically supported work-procedures in form of diagrams and to discuss and plan the incorporation of a CSCW-system into work procedures based on these diagrams.

The discussion of the technically supported work-procedures is oriented along the activities which are displayed and discussed sequentially, step by step. Such a workshop design puts the focus on the context of use rather than on technical details; hence the name „socio-technical walkthrough". Before the workshop design STWT is explained in more detail, a summary of its history is provided.

STWT – History

The workshop design STWT is the result of a methodological reflection of about 8 projects conducted in my workgroup [Herrmann, Kunau, Loser (2002b)] [26]. In all projects SeeMe diagrams were used for supporting organizations in appropriating IT-systems. Although no consistent strategy for facilitating workshops was pursued, there were best practices that could be identified in an ex-post analysis. The result is a generic workshop design that can be applied in the context of CSCW-projects whenever computer supported work procedures have to be discussed.

[26] Insights from early phases of case study 1 are already included in the initial concept of the STWT and have been published in more detail in [Herrmann et al (2004b)].

After this initial publication further research was invested in elaborating the facilitation concepts. Both of my case studies were part of this research; three strands of research can be identified:

- Detailed facilitation techniques: Loser elaborated and evaluated detailed facilitation techniques to be employed when supporting group discussions with semi-structured diagrams [Loser (2005)]. These include sketching and correcting of diagrams in workshops, aesthetical improvement of diagrams created during workshops, transfer of diagrams to other media, and presentation of diagrams to a group. The workshops conducted during the case studies 'Mobile communication system' (cf. section 6.2) and 'Electronic ToC-service' (cf. section 6.3) employed Loser's concepts but also contributed to their further development. Especially the concepts of the guiding question (cf. section 7.3.1, p. 225) and letting small groups prepare diagrams (cf. section 7.3.1, p. 232) were conceived and evaluated within the research presented in this thesis.
- SeeMe Editor: In the early case studies, SeeMe diagrams were drawn using conventional means such as card boards, cards and markers. Based on the experiences an editor was designed and implemented [Loser (2005), section 9]. The two case studies presented in this thesis were the first to employ the SeeMe editor in workshop settings. Thus all results pertaining to the technical support for STWT workshops originate from the research presented in this thesis.
- Integration with software development: The early case studies concentrated on requirement-elicitation on the one hand and deployment of completed software on the other. The two case studies presented in this thesis were the first to employ the STWT in the context of all phases of software engineering between requirements elicitation, deployment and use. Thus questions related to the integration of issues of software development and organizational change within one workshop originate from this thesis.

The following sections describe the basic concepts of the STWT as they have been employed in the case studies; they serve as a baseline for analyzing the case studies (cf. chapter 7) and for deriving further recommendations for the framework presented in chapter 8.

Facilitated Workshops – Basics

Since the term workshop is often used in a very broad sense, some clarification is needed. I use the term "workshop" to denote professionally organized meetings during which the participants elaborate a result. The methodological support for group discussions is a profession of its own right performed by facilitators [Webpage IAF]. The role of a facilitator can be described as follows [HL&D (2002), p. 6]:

> *„A facilitator is someone who uses knowledge of group processes to formulate and deliver the needed structure for meeting interactions to be effective. The facilitator focuses on effective processes (meeting dynamics) allowing the participants to focus on the content or the substance of their work together."*

I prefer the term „workshop" to the term „meeting" for two reasons. First, because it relates to other concepts in the field of CSCW such as a „future workshop" [Ehn, Sjögren (1991)] or JAD-workshops ([Crawford (1994)], cf. section 4.3.3, p. 115). Second, because the word „work" has the connotation of elaborating some result.

The definition of a facilitator given above includes the task of delivering *„the needed structure for meeting interactions to be effective"*. For this purpose templates for structuring workshops are elaborated and published. In the field of systemic intervention (cf. section 4.2) such templates are referred to as „intervention design" [Königswieser, Exner (2004), p. 30]. In analogy templates for workshops in general are referred to as workshop designs. Workshop designs are templates either for a complete workshop or for parts of it [Schuman (2005)]. A typical storyboard for a workshop during which the participants need to come to a conclusion on some topic consists of the following phases [Klebert, Schrader, Straub (1987), pp. 125]:

- Reception, opening, warm-up: The initial phase is important for creating an open and communicative atmosphere.
- Orientation towards topic/problem: The goal of this phase is to create awareness for the topic/problem and to prepare the next phase e.g. by dividing the overall topic/problem into smaller parts.
- Elaboration on topic/problem: During this phase the actual problem solving takes place. Depending on the size of the group, the participants break out into smaller subgroups to work on parts of the overall topic/problem.
- Integration and concept for solution: The results of the small groups are presented for the whole group.
- Actions: An action list is created to ensure that the results of the workshop are put into practice.
- Closing: A workshop should have a defined ending during which the result, the process and the emotions connected to the workshop can be reflected.

When a facilitator prepares a workshop he often fills in a matrix like the one in Table 13; for each phase the five aspects goal, technique, resources, responsibility and time are detailed. The sheet is then taken into the workshop as guidance. For most STWT-workshops conducted during the case studies such sheets are available for analysis (cf. appendix 11.1).

	Goal	**Technique**	**Resources**	**Responsible**	**Time**
Reception / Opening					
Warm-up					
Orientation					
Elaboration					
Integration					
Actions					
Closing					

Table 13: Matrix for Workshop Preparation

The workshop design STWT should be used by facilitators for facilitating workshops in the context of CSCW-projects during which computer supported work procedures are discussed. The following sections discuss in detail how the STWT extends the general template for workshops.

STWT Topic: Computer Supported Cooperative Work Procedures

STWT workshops always address the topic of computer supported cooperative work procedures within an organization. All further techniques included in the workshop design are directed towards this purpose. STWT-workshops may be conducted during all phases of a CSCW-project and they may serve different goals within the projects, but they are always related to CSCW. Examples for different STWT-sessions that are included in the case studies are: documentation of status quo at the beginning of a project; elaborating requirements for the CSCW-system; validating a prototype of the CSCW-system; training; discussion of conventions for use of CSCW-system.

STWT: Walk-Through Concept with Guiding Questions

The basic idea of the STWT as a workshop design is to walk through semi-structured diagrams that represent an organizations (future) computer supported work procedures step by step. For each step a guiding question is discussed and changes are applied to the diagram as needed.

Thus the STWT bears relations to other walk-through techniques known in computer science: code walkthroughs which are used for validating the implementation of a software design; the cognitive walkthrough which is an inspection-based method for usability testing [Polson et al. (1992)]; or its extension, the groupware-walkthrough [Pinelle, Gutwin (2002)] which takes into account the cooperation of several users.

One of the basic concepts of walk-throughs is that the artifact in question is checked according to a set of predefined questions. For the cognitive walkthrough [Cockton, Lavery, Woolrych (2003), p. 1124] give examples of questions that are to be answered for each part of the GUI: „Will the user try to achieve the right effect?”, „Will the user notice that the correct action is available?”, „Will the user associate the correct action with the effect that the user is trying to achieve?”, „If the correct action is performed, will the user see that progress is being made toward the solution of the task?”. In [Herrmann, Kunau, Loser (2002b)] and in [Loser (2005), p. 186] the question suggested for the STWT is „What would happen if the illustration became reality?”. This question should support the participants in imagining their future computer supported work procedures, discussing the consequences and establishing plans and conventions.

STWT: Diagrams as Resources

The STWT makes use of semi-structured diagrams as a resource in most phases of a workshop; they are used for documentation, visualization and as guiding artifacts. All three aspects shall be elaborated briefly:

Documentation: One purpose of STWT-workshops is to support the members of an organization in elaborating a socio-technical self-description. In section 2.4.2 it is discussed that modern organizations need to make their self-descriptions durable by writing them down. In STWT-workshops semi-structured diagrams are used for documenting the socio-technical self-descriptions evolving during the workshops. This means that decisions about the future computer supported work procedures are expressed in terms of SeeMe elements and stored as semi-structured diagrams. Conventional means for documentation such as notes on flipcharts or cards on pin boards will still be used, but the main form of documentation consists of semi-structured diagrams.

Visualization: it is a central aspect of facilitation that much information is visualized for all participants. This applies to decisions as well as to open issues or organizational aspects such as the agenda [Klebert, Schrader, Straub (1987), pp. 119]. Conventional facilitation techniques foresee posters, flipcharts and pin boards; these are hung up around the room such that they are visible to all participants at all times. This is an important element for facilitation because it keeps the issues of the workshop more tangible. During an STWT most issues are documented in form of semi-structured diagrams which are edited using the SeeMe-editor. This implies that the diagrams also largely replace the other forms of visualization.

Guiding artifact: within the socio-technical walkthrough the diagrams serve as guiding artifacts. They are used as guideline for facilitation to ensure that both organizational and technical aspects are discussed in a systematic manner. In other walkthroughs the object of evaluation is directly available. Code walkthroughs go through each line of code step by step; the cognitive walkthrough takes the graphical user interface and steps through each mask. The socio-technical walkthrough is less immediate. The object of inspection and design are the semi-structured diagrams that represent the future computer-supported work-procedures.

STWT: Tool for Presenting, Discussing and Editing Diagrams

Based on the experiences of earlier case studies [Herrmann, Kunau, Loser (2002b)] the SeeMe-Editor[27] is being developed in my workgroup. It combines features for drawing and editing semi-structured diagrams with features for their presentation and discussion in workshops [Loser (2005), chapter 9]. The following is a brief description of those features that are of special relevance.

- Support of SeeMe-Notation: the editor offers elements and functions necessary for creating diagrams according to the SeeMe syntax (cf. section 4.3.5, p. 126; section 5.2.2, pp. 139; appendix 11.2).
- Style guide for ergonomic layout of diagrams: since SeeMe-diagrams do not have a predefined layout (such as “swim lanes” which are often suggested for EPC or activity diagrams in UML [Rumbaugh, Jacobson, Booch (2005), p. 97]), layout rules have been developed and implemented in a critiquing component for the editor [Wacker (2000)].
- Hide and show: especially during group discussions it is distracting to display a complex diagram in one block. The SeeMe-editor offers special features for blinding out parts of a diagram and then displaying them again. The hide and show mechanisms are available either for spontaneous use during a discussion or for preparing predefined sequences – so-called snapshots – to be used during a presentation. The hide and show mechanisms were extensively used within the STWT-workshops of the case studies; the usage of diagrams as guiding artifacts based largely on the feature of sequentially displaying elements from a diagram (cf. section7.3.1).

[27] Information and free download: [Webpage SeeMe Editor]

- Integration of additional media: the SeeMe-editor allows adding hyperlinks at any place in a SeeMe-diagram. Via HTTP other applications can be started for displaying additional media. This feature is used in the case studies for integrating the representation of work procedures with artifacts such as software-specifications, prototypes or screenshots (cf. section 7.2.2).

In both case studies the SeeMe-editor is used for creating, presenting and editing all diagrams.

Workshop-Related Research Questions for the Case Studies

For the workshops the facilitation method of the socio-technical walkthrough is employed. One aspect for the analysis is to find out how this concept worked in the context of software-development. The analysis concentrates on the function of the semi-structured diagrams. Could they fulfill their multiple roles of documenting the organization's StSd, visualizing the discussion, and serving as a guiding artifact for the facilitation of the workshops? Since the SeeMe-editor was used for handling all diagrams, the question related to the diagrams cannot be separated from the evaluation of the tool support. Therefore the first workshop-related research question asks for the experience with semi-structured, electronic diagrams as guiding artifacts in STWT-workshops (WS1 in Table 14).

The analysis in chapter 7 is based on examples using the SeeMe-editor, but the conclusions drawn from the empirical evidence are of general nature giving advice for the usage of editors within workshop settings. This seems to be a relevant issue, because diagrams are frequently used in group discussions and editors exist providing special features for different purposes [Fank (1998)]. But experiences with using such editors in facilitated group discussion are not yet well documented and examined.

The next research question concerns the integration of technical and organizational topics within the workshops. Sections 4.1 and 4.2 discuss potential problems in the integration of software engineering and systemic intervention; it is reasonable to assume that such problems would become apparent during the workshops when technical as well as organizational issues are discussed. Research question WS2 therefore asks for the experience made with the integration of technical and organizational issues in the STWT-workshops.

5.2.4 StSd – Integrating SWE and Organizational Change

The methodological support for StSd discussed so far concentrated on adding support for structural coupling to a CSCW-project. This is important because the discussions in chapter 4 identified a lack of support here. But how is this support included in an overall project design?

The methodological support for socio-technical self-description is integrated into the project model for CSCW-projects I elaborated in section 3.2.2 (cf. Figure 11). In this project model I describe the outcome of any CSCW-project as a socio-technical setting (cf. Term 22), which is bipartite comprising a socio-technical organization (cf. Term 20) and a CSCW-system (cf. Term 7). To achieve this outcome, the people responsible for a CSCW-project need to be bound to a socio-technical perspective throughout the project; i.e. they need to keep the awareness for the bipartite goal alive. It is one of the research questions for the case studies how this can be done. I suggest using the methodological components of semi-structured diagrams for documenting the organization's socio-technical self-description and STWT-workshops for creating, discussing and maintaining them.

The STWT-workshops should provide a setting for the integrated discussion of technical and organizational aspects. The facilitator is responsible for ensuring that both, organizational and technical topics are brought up and discussed. In particular he needs to make sure that organizational implications of design decisions are discussed and documented. For documentation, the semi-structured diagrams created as the organization's self-description should be used.

Here the role of socio-technical self-descriptions as boundary objects becomes relevant (cf. section 3.1.3; p. 72). The idea is, that the StSd "sits in the middle" [Star (1989), p. 46] and serve as boundary objects between different groups of users; between users and management; between users and software-developers etc. They should be a guiding artifact throughout the whole project; just as it is suggested for use cases (cf. section 4.1.5, p. 99) or scenarios (cf. section 4.3.3, p. 120). But by documenting social as well as technical issues they should be able to support the integration of software engineering and organizational change on project level; and by documenting conventions they should be able to describe the identity of the socio-technical organization that makes use of the CSCW-application.

Research Question for the Case Studies

There is no reported experience with keeping a socio-technical perspective throughout all phases of a CSCW-project. The case studies will have to show whether this is possible, how the methodological components can be used to keep this perspective alive, and what the outcome of such a CSCW-project will be.

Research question W2 which is derived in the previous section addresses issues related to integrating technical and organizational issues within one workshop. Another open issue is, whether the semi-structured diagrams can indeed function as boundary objects on project level between the heterogeneous groups involved. Research question I1 (cf. Table 14) is included into the analysis for providing answers.

The final research question then is, whether the overall concept was successful. Did we succeed in creating socio-technical settings at the end of the projects? Research question I2 addresses this issue.

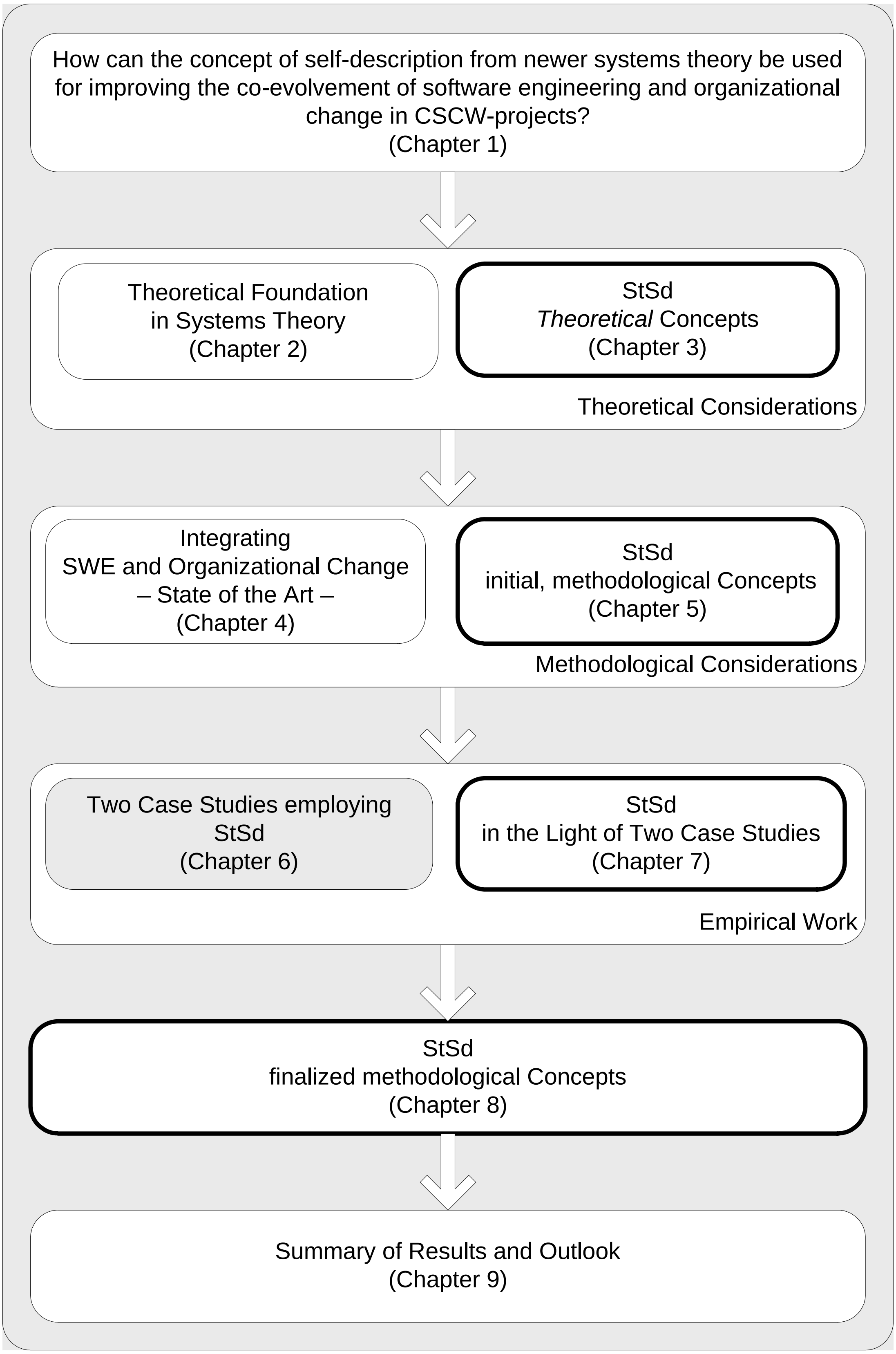
5.3 Summary of Research Questions

To conclude this section, Table 14 summarizes the research questions for the analysis of the case studies.

Id	Research Question	cf. Section
Content of Socio-technical Self-descriptions (StSd)		**7.1**
C1	What are the contents of documents that contribute to the socio-technical self-description of the socio-technical organization with regard to the new CSCW-system?	7.1.1
C2	Do the topics identified in C1 differ from the scope of other documentation such as use-cases, business-models or scenarios?	7.1.2
C3	How did the topics that were eventually documented emerge during the project?	7.1.3
C4	What importance did the participants attach to the contents of socio-technical self-descriptions?	7.1.4
Form of Socio-technical Self-descriptions (StSd)		**7.2**
F1	Were semi-structured diagrams suitable for documenting the topics discussed during the project?	7.2.1
F2	How is the software-system represented in the socio-technical diagrams and how does this representation change in the course of the project?	7.2.2
Facilitated Workshops		**7.3**
WS1	What is the experience with electronic, semi-structured diagrams as guiding artifacts for the facilitation of workshops?	7.3.1
WS2	What was the experience with integrating organizational and technical issues within the workshops?	7.3.2
Integration of SWE and Organizational Change		**7.4**
I1	Can artifacts of socio-technical self-description relate the two processes of software engineering and organizational change to each other by serving as boundary objects?	7.4.1
I2	Were the CSCW-projects successful in creating socio-technical settings?	7.4.2

Table 14: Research Question for the Analysis of the Case Studies

The following chapter 6 will introduce the case studies before chapter 7 discusses each question of Table 14 by drawing on the experience of the case studies. The rightmost column of Table 14 contains the section in which the corresponding question is discussed.


How can the concept of self-description from newer systems theory be used for improving the co-evolvement of software engineering and organizational change in CSCW-projects?
(Chapter 1)
Theoretical Foundation in Systems Theory
(Chapter 2)
StSd
Theoretical Concepts
(Chapter 3)
Theoretical Considerations
Integrating SWE and Organizational Change
– State of the Art –
(Chapter 4)
StSd
initial, methodological Concepts
(Chapter 5)
Methodological Considerations
Two Case Studies employing StSd
(Chapter 6)
StSd
in the Light of Two Case Studies
(Chapter 7)
Empirical Work
StSd
finalized methodological Concepts
(Chapter 8)
Summary of Results and Outlook
(Chapter 9)

6 Two Case Studies employing StSd

I conducted two case studies which explore my methodological concepts elaborated in the previous chapter 5. This chapter contains descriptions of the research method (section 6.1) and the two case studies "mobile communication system" (section 6.2) and "electronic ToC service" (section 6.3). Section 6.4 provides an overview of the empirical data that is available for analysis, and discusses the relative strength and weaknesses of both case studies.

6.1 Research Method: Action Research

6.1.1 Introduction into Action Research (AR)

The research method applied is action research (AR) which has a long tradition in social science and especially in projects concerned with organizational change. What are the characteristics of action research? A definition that already integrates several historical strands is the following [Hult, Lennung (1980), p. 247]:

> *„Action research simultaneously assists in practical problem-solving and expands scientific knowledge, as well as enhances the competencies of the respective actors, being performed collaboratively in an immediate situation using data feedback in a cyclical process aiming at an increased understanding of a given social situation, primarily applicable for the understanding of change processes in social systems and undertaken within a mutually acceptable ethical framework."*

The key aspect of AR is that research and actual problem solving in an organizational situation coincide. Originating in social sciences, action research has been transferred into the field of CSCW. Several of the methods discussed in section 4.3.3 were developed, evaluated and improved using action research. For MUST the authors state that it *„has been developed throughout 10 projects in Danish and American organizations"* [Kensing, Simonsen, Bødker (1998), p. 168]; SSM was subject to evaluation and improvement in action research projects [Checkland (1999), pp. 146]. Without explicitly naming the method of action research, Mumford applies its general principles. In [Mumford (1995)] she describes her method QUICKethics followed by three case studies [ibid. pp. 134]. For each case study she provides conclusions in which she reflects on the usage of her method ([ibid. p. 143], [ibid. p. 154], [ibid. p. 162]). In

[Avison, Wood-Harper (1990), p. 180] the authors describe how they used action research to evaluate and improve the methodological framework of Multiview.

6.1.2 Application of AR in Case Studies

Action research was chosen for two reasons. First, AR is suitable for achieving the purpose of the case studies, which is to provide further insight about the usefulness of socio-technical self-descriptions in CSCW-projects. The initial design of methodological support (cf. section 5) incorporates many theoretical ideas but is still quite vague concerning the practical realization. It is an appropriate methodological complementation at this point, to practically employ the concepts and learn from the experience.

Category in Action Research	**Concretion in Case Studies**
Theoretical Framework (F)	Systems theory; concepts of socio-technical organization and socio-technical self-description (StSd)
Research Method (M_R)	Action Research
Problem Situation from researcher's point of view (A)	How can artifacts of socio-technical self-description be integrated into CSCW-projects that include the design and implementation of a new CSCW-system. The research questions detailing this general problem situation are deducted in sections 5.2.2, 5.2.3, and 5.2.4; in addition they are summarized in Table 14.
Problem Situation in which researchers are intervening (P)	
P1 "Mobile communication system"	Designing and testing new software and new computer-supported work-procedures in the context of e-business.
P2 "Electronic ToC-service"	Designing and testing new software and new computer-supported work-procedures for using an electronic ToC-service for scientific periodicals.
Problem Solving Method (M_{PS})	Creation and usage of StSd as described in the initial concept in chapter 5

Table 15: Case Studies in Terms of Action Research

Second, the first case study 'mobile communication system' was part of a publicly funded project from which the German government requested both: an increase in scientific knowledge and practical, local problem [BMBF (2002)]. AR is a method offering exactly this combination. The second case study „electronic ToC-system" takes place in my own work group, and I am directly involved in the project. Thus the setting of the second case studies lends itself to action research very well.

There are objections against action research: In [McKay, Marshall (2001)] an overview of the usage of action research within the field of information system research (IS research) is given along with criticism of the sometimes unsystematic use. The authors emphasize that although research and problem solving take place at the same time and in the same setting, the issues must be kept apart analytically. For this purpose they distinguish two cycles within AR projects: one for problem solving and one for research. For each of these cycles the problem and the corresponding methods have to be defined. With reference to the scheme described by [McKay, Marshall (2001), p. 56] the setting for the two case studies are summarized in Table 15. The items F, M_R, and A describe the research interest while the items P and M_{PS} describe the practical problem to be solved.

6.1.3 Method within the Research Cycle: Grounded Theory

The research cycle in this thesis is of exploratory nature. The first goal is to find out whether it is possible to transfer theoretical ideas on StSd into practice by employing the methodological concepts described in chapter 5. The description of the case studies in this chapter should elucidate how they were put into practice. The subsequent analysis in section 7 will reason about the course of events in the case studies with the goal to come to explanations and new ideas that will contribute to a refined approach. This refined approach is the second goal of the case-studies and will be described as a guiding framework in chapter 8.

Exploratory case-studies do not allow any comparative statements about the usage of StSd such as „using StSd will increment the number of software-functions adopted by the users by 30%". The concept is too new and therefore still too vague to formulate appropriate hypotheses and to design studies featuring control groups. The strength of exploratory case-studies is to provide insight into an approach that exceeds those that can be derived from theoretical considerations. With regard to StSd the analysis of the case-studies should lead to a more detailed description of a methodological framework integrating software engineering and organizational change in a CSCW-project.

But how are the case studies analyzed? Principles described in [Strauss, Corbin (1998)] for developing grounded theory, are employed. The basic idea of developing grounded theory is that the data collected during field work is used to develop a theory „grounded" in the data. There is no hypothesis before hand that is to be confirmed or rejected by the data. This approach matches the situation for the case studies very well, as there is no hypothesis to be confirmed; rather empirical experience is needed to add to the theoretical insights for developing a methodical framework.

In both case studies data was collected in form of audio- and video recordings of workshops, preparatory documents of workshops, minutes of meetings, photos etc. (details of the empirical material are given in section 6.4.2, pp. 164). As suggested for developing grounded theory [ibid. pp. 40], the analysis of the data is guided by research questions. Their deduction is included in the description of the initial concept in chapter 5 and summarized in Table 15 and Table 14.

Again, as pointed out in [Strauss, Corbin (1998)] it is nearly impossible and not even helpful to analyze the full amount of empirical material on the level that is called „microanalysis" in [ibid. pp. 57]. Rather the microanalysis is used on a part of the material as a starting point; then a technique called theoretical sampling is used. In the qualitative research described by Strauss and Corbin, theoretical sampling guides the further gathering of data. Based on intermediate research results, the researcher decides on further data to be selected (e.g. more interviews to be conducted, more observations to be made). Since my case studies are action research projects, I do not visit the projects for research purposes only; I am involved in the ongoing project activities anyway. Therefore I collected data whenever I was doing work for the projects and made use of this abundant resource as proved necessary during the analysis. But also, as originally intended by the technique of theoretical sampling, the experiences made during the projects lead to the decision that more or other data was needed. As a result questions were added to existing questionnaires (e.g. those described in section 11.5.1) or new questionnaires were derived (e.g. the questionnaires distributed in case study 2, cf. section 11.5.2). Thus for my case studies, theoretical sampling includes both: selection from existing material and gathering of new data.

The employment of action research leads to one further addition to the original concept of qualitative research as suggested by Strauss and Corbin: while their concepts work on (textual) data that are gathered by the researcher, as an action researcher I dispose of yet another resource – my own experience. Since I am part of the projects, I directly experience the effects of my own doing. With respect to scientific research, this is a double-edged sword: on the one hand I gain insights that would otherwise be disclosed to me; on the other hand it is difficult to maintain the objective distance necessary for analysis. To cope with this problem, I kept a research diary into which I wrote any experiences made during the projects that seemed important. The content of this research diary was then used as one source of (textual) data during the analysis. This way of proceeding cannot completely solve the general conflict, but it does make central aspects of my personal insights available for analysis by others.

As foreseen by Strauss and Corbin [ibid, pp. 57] I used a form of microanalysis to start the analysis of the case studies: the contents of the diagrams were viewed and categorized (cf. section 7.1.1). From there on – guided by the research questions – I derive the empirical insights for the usage of StSd in continuous process: The evidence found in the data is compared to other data from the case studies as well as to findings in literature [ibid, pp. 48]. These comparisons lead to further insights that are again backed by the data from the case studies. Chapter 7 documents this process of analysis, while chapter 8 presents the results in form of a framework for using StSd in CSCW-projects. But first, both case studies are presented such that the reader can gain a general idea of their settings and contents.

6.2 Case Study 1: Mobile Communication System

6.2.1 Topic and General Description

Work-Procedures

In the first case study, a company plans to improve its communication processes by using mobile information technology. Hence the case study will be called „Mobile Communication System" throughout this text. The company offers logistics services and intended to support the interaction between the drivers on the road and the dispatchers in their offices and thereby improve their business processes. Figure 20 shows pictures of the dispatchers' and drivers' workplaces. The unit of the company that is involved in the case study is called „Westkreis". „Westkreis" is a team consisting of a team-leader who has also worked as a dispatcher, seven dispatchers working in three offices in three different towns and 17 drivers. In addition two managers of the logistic-company's head office have advisory functions for the team. „Westkreis" is responsible for the complete logistics of delivery of a large steel trading company that had outsourced this division: The dispatchers of „Westkreis" receive purchase orders from the steel trading company and sort them out into delivery-tours for the drivers. The drivers load their trucks according to the pile of purchase orders they receive and are on tour for between 4 and 10 hours a day. During the tour they only communicate with the dispatchers if anything irregular happens. Early in the morning and in the evening, before and after their daily tour, the drivers come into the office to hand in the documentation of their last tour and receive the paperwork and additional information for their next tour.

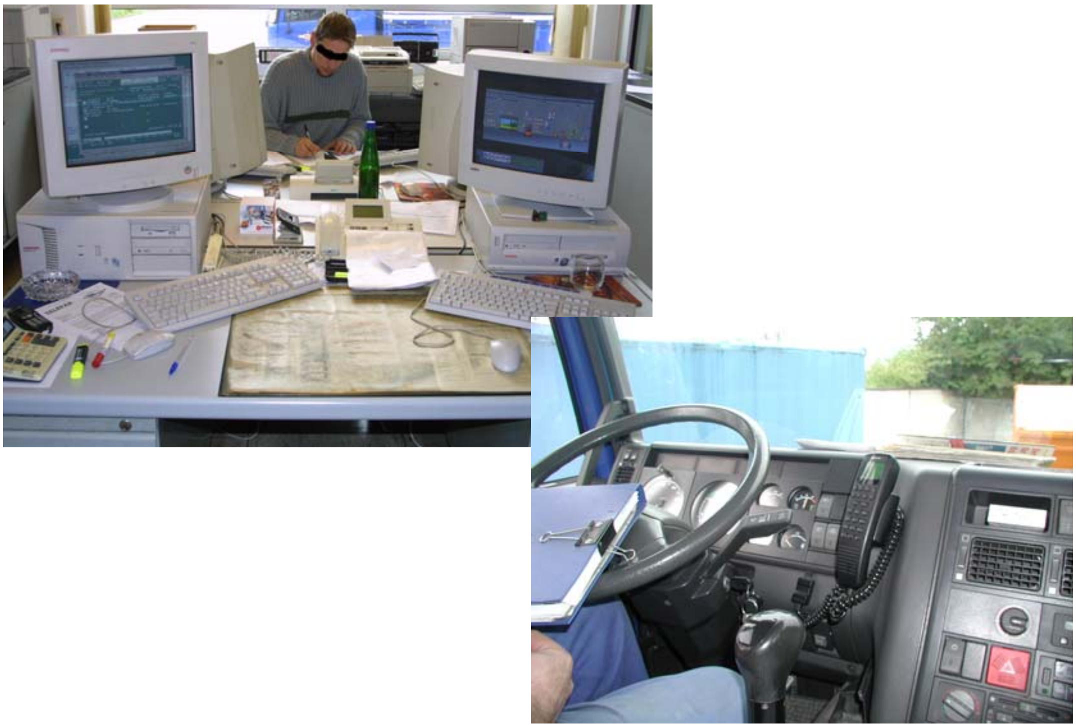


Figure 20: Workplaces in 'Westkreis'

The management of „Westkreis" phrased their expectations to the new IT-system as follows:

- on time information about the status of all tours for the dispatchers
- on time information about new tours for the drivers
- integration of the drivers into the business processes
- reduction of paperwork

Characteristics of Cooperative Work

Term 3 defines characteristics of cooperative work; a brief review how these characteristics manifest themselves in the work-procedures in „Mobile Communication System" is inserted as this point.

The interdependent actors in this case study are the drivers and the dispatchers; the common field of work that they share consists of the delivery tours that are planned by the dispatcher and driven by the driver. The driver's individual field of work comprises the list of jobs he has to do, the truck which he loads and unloads and the delivery notes that the recipients sign upon acceptance of the goods. Whenever the driver completes a delivery he changes his individual field of work: the truck carries fewer goods, the delivery notes are signed and the list of open jobs is shorter. The dispatcher's individual field of work comprises all orders that need to be delivered, the material available in the hall, the drivers and trucks available for tours, the papers documenting the tours for triggering further administrative processes. The driver's activities influence the dispatcher's field of work by being available or unavailable for further tours and by changing the state of orders to „delivered". The dispatcher's activities influence the driver's work by assigning new orders which have to be integrated into tours.

Context of Research Program

The case study is embedded into the research project SpiW[28] which is part of „work in the e-business" [BMBF (2002)], a research program of the German government. One goal of this research program was to describe the effects of different forms of electronic business support on the organization of companies and work arrangements; another to develop and try out methods to shape working processes and conditions in the e-business. The overall goal of SpiW-Com was to design, implement and test a technical infrastructure and corresponding work procedures to support the communication between dispatchers and drivers. The working name for the IT-system was SpiW-Com.

6.2.2 IT-System: SpiW-Com

General Description

The architecture of the IT-system SpiW-Com is described in [Gruhn et al (2003)]. SpiW-Com consists of three components: the mobile devices for the driver, stationary devices for the dispatchers' workplace and an application server. Figure 21 illustrates the architecture of SpiW-COM. While the mobile devices communicate with the application server using wireless telecommunication services such as GSM, GPRS or UMTS the stationary devices are connected to the application server via LAN. Based on client-server architecture, the application server contains the business logic implemented as workflows. Both the mobile client and the dispatcher's software are implemented as „thick" clients, i.e. containing parts of the business logic, but for different reasons. The mobile client needs to function independently even if its connection to the application server is temporarily lost. The dispatcher's software needs the complete logic for handling the data. Thus there are local workflows on both clients. Clients and server use XML-based SOAP for structured data exchange.

[28] German acronym for "Speditionen im Web" which refers to the usage of web technology within logistic companies

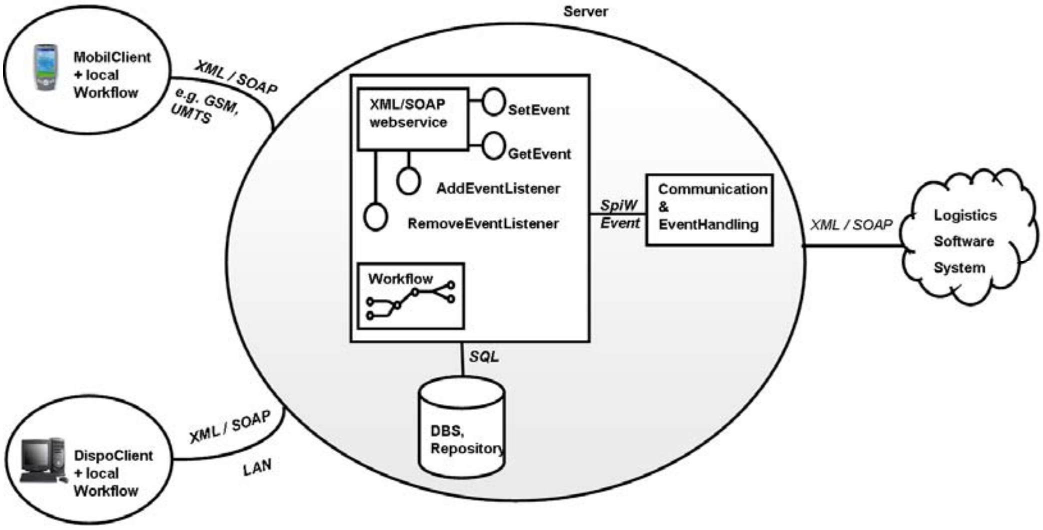


Figure 21: Architecture of SpiW-Com
[Gruhn et al (2003)]

6.2.3 Interventions and Results

Research groups from three disciplines cooperated in the project SpiW; all three contributed to the development of the IT-system but each one had a special focus determined by the disciplinary background:

1. Software engineering: their focus was on using and evaluating new technologies for the design and implementation of IT-infrastructure for mobile e-business processes.
2. Logistics: their focus was to evaluate and design the changes in business processes that were caused and enabled by the new technical infrastructure.
3. Informatics & society: my workgroup was responsible for evaluating the changes which the new technical infrastructure would cause in the working condition, and for the methodical support of shaping new technically supported work-procedures.

The last aspect „methodical support for shaping new technically supported work-procedures" was my subproject within the overall project. Based on the initial concept described in section 5.2.3, I planned, organized, conducted and evaluated workshops with members of the „Westkreis". In all workshops, semi-structured diagrams were used to discuss and plan the future technically supported work-procedures. Section 11.1.1 and 11.1.2 in the appendix (pp. 299) contain a detailed list of the interventions in the case study; section 11.3.1 (pp. 308) contains a representative set of diagrams created during the case study.

The project ended with a qualification workshop during which individual training of the drivers was combined with a group discussion on conventions needed for using the CSCW-system SpiW-Com. During this workshop the completed version of the CSCW-system SpiW-Com was used. So the project yielded a CSCW-system named SpiW-Com and related socio-technical self-descriptions of the logistics company. Due to its characteristic of being a research prototype, the system SpiW-Com never became operational, it was not possible to evaluate whether a socio-technical organization in the sense of Term 20 evolved as a result of the project.

6.3 Case Study 2: Electronic TOC-Service for Scientific Periodicals

6.3.1 Topic and General Description

Work-Procedures

The goal of the project of the second case study is to replace a paper-based procedure for ordering articles from scientific periodicals by an electronically supported procedure. This case study will be referred to by „Electronic ToC-Service" throughout this thesis.

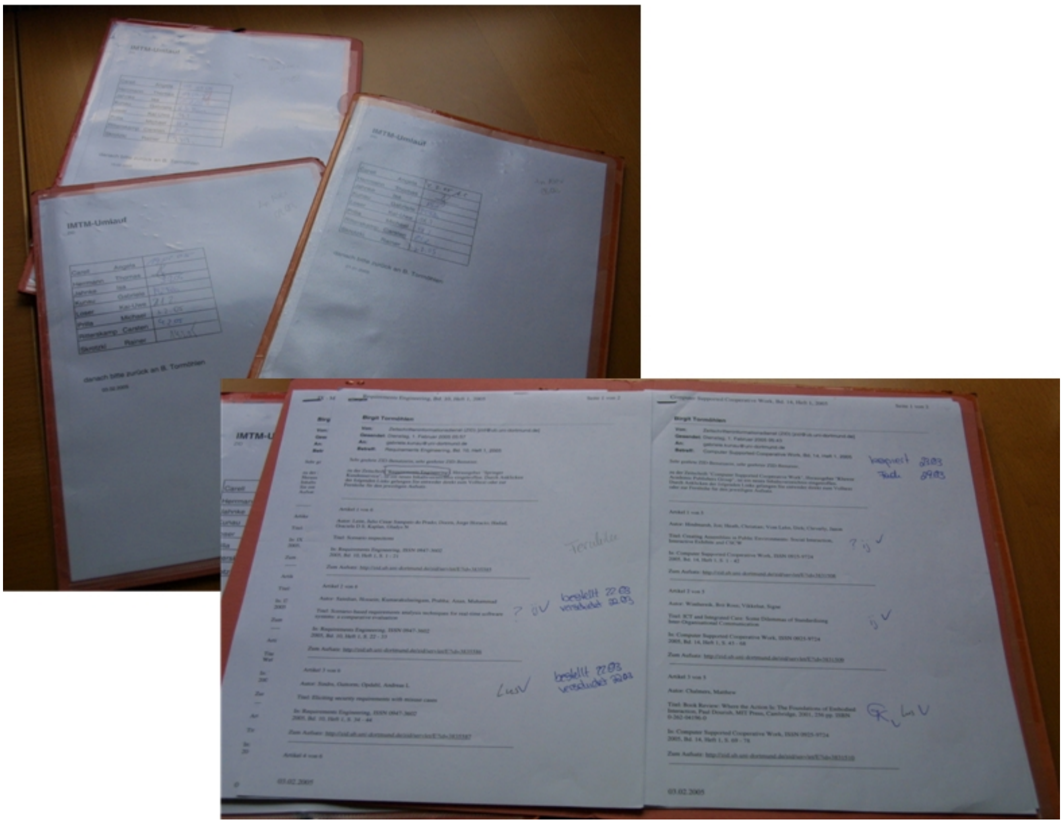


Figure 22: Picture of Circulating Files with ToC

The project is situated in a German university. The university's library subscribed centrally to important scientific periodicals. Whenever a new journal arrived, copies of its table of contents were made and distributed via internal mail to those research groups who had assigned for the service. There the tables of contents were collected in the secretary's office and every few days a file with 2 to 10 tables of contents was created and circulation among the members of the group was initiated. Figure 22 provides a picture of such files. Whenever a researcher found a file in his post-box, she would read the table of contents, mark the articles she was interested in and pass it on to the next researcher. At the end, after all researchers had read and marked the tables of contents, the file was given to a student worker who went to the library, Xeroxed the marked articles and handed them over to the researchers to read and use in their work.

In 2003 the university's library started a new service called ZID ("Zeitschrifteninformationsdienst", the literal translation would be "journal information service"). The contents of new journals are sent out by e-mail directly to each individual researcher who subscribes to the service. For each article the mails contain a link to the abstract or even the full text depending on the library's agreements with the publisher. The library's goal was clearly to abolish the task of producing and distributing the copies of the table of contents.

My workgroup wanted to make use of the new service, but also saw negative effects that needed to be avoided. The file, circulating through the group, fulfilled tasks of coordinating the group's way of handling scientific literature. Notes and recommendations were added to the papers, and everybody was interested in the orders the colleagues placed. The head of the team even admitted that he would mark articles in the name of others if thought that the person should read the article. In the first workshop conducted in the course of the project the advantages and disadvantages of electronic table of contents were listed by the participants; they are summarized in Table 16.

The goal of the project within our workgroup was to design and implement a CSCW-application and to organize procedures that made use of the new service from the library while avoiding the disadvantages listed in Table 16.

Advantages	**Disadvantages**
The individual researcher is independent from the circulating file	Readability of paper is better; it is faster to read
Notes about the reasons for ordering an article can be added	Many mails will be generated rather than marking an article on one piece of paper
Fewer articles will be ordered (because of the chance to view the abstract)	Coordination is necessary when several researchers order the same article
Overview of orders can be generated	Danger that mails are just „clicked away"
Less effort	Less social pressure to acknowledge the information
Individual flexibility regarding timing	No information about the colleagues' interests
	Less information on a casual way

Table 16: Advantages and Disadvantages of an Electronic Toc-Service

Characteristics of Cooperative Work

The characteristics of cooperative work as defined by Term 3 shall be reviewed briefly at this point. The work-procedures in the case study „Electronic ToC-Service" include two cooperative relationships:

a. The scientists cooperate with each other by selecting and recommending articles to read.

b. The scientists cooperate with the student workers in the library by reading tables of content and placing orders for articles.

Therefore the interdependent actors are the individual scientists and the student workers in the library. The common field of work consists of articles in scientific periodicals which need to be acknowledged and read by the scientists and which need to be acquired and archived by the student workers.

The scientists' individual fields of work include their lists of new tables of contents and open orders. By reading tables of content and recommending articles, the scientists make changes to each others lists. By reading and ordering articles, they make changes to the libraries field of work. Here lists of new and open orders make up the individual field of work. Once the library team has retrieved an article and closed an order, the researcher's field of work is affected again, because now he has the article on his desk and needs to decide whether it should be included in the teams literature database or not.

6.3.2 IT-System: ELISE

The IT-system designed and implemented is named ELISE (German acronym for electronic ToC-system). I implemented the current version of ELISE based on a first prototype designed and partially implemented by Ho [Ho (2004)]. Figure 23 illustrates the architecture.

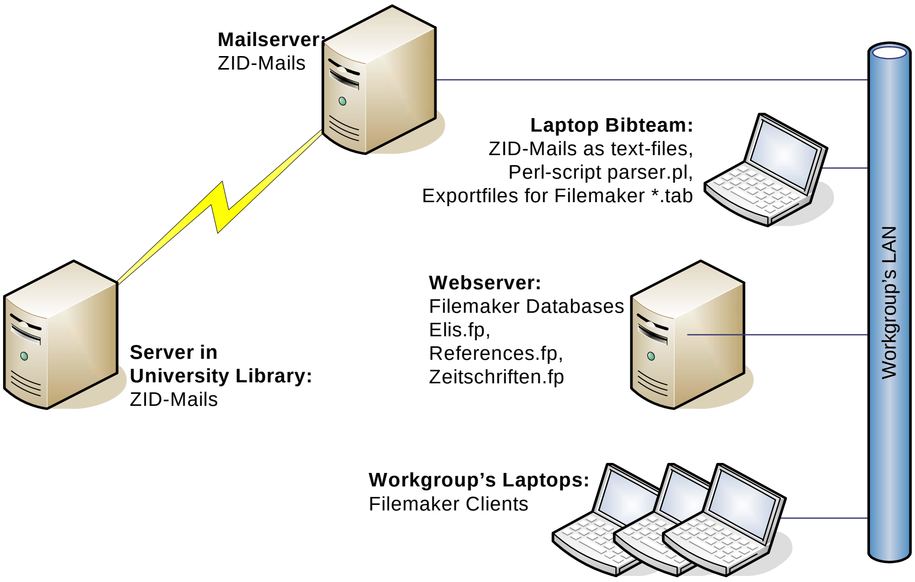


Figure 23: Architecture of ELISE

The ZID-Mails are sent by a server of the university's library and received by the workgroup's mail server. From there they are read by the library team and stored in form of text files. A PERL-script is used to transform the ZID-Mails into tabular data that can be imported by databases configured with Filemaker®. An automatic ingest of ZID-mails directly by the PERL-script is possible, however no reliable syntax specification for these mails was available and manual quality assurance is necessary to cope with variances in the mails structure. The Filemaker® databases are located on the workgroup's web server from where they can be accessed by all members of the workgroup. There are three interrelated databases:

a) Elis.fp stores information related to table of contents (i.e. to issues of a journal).

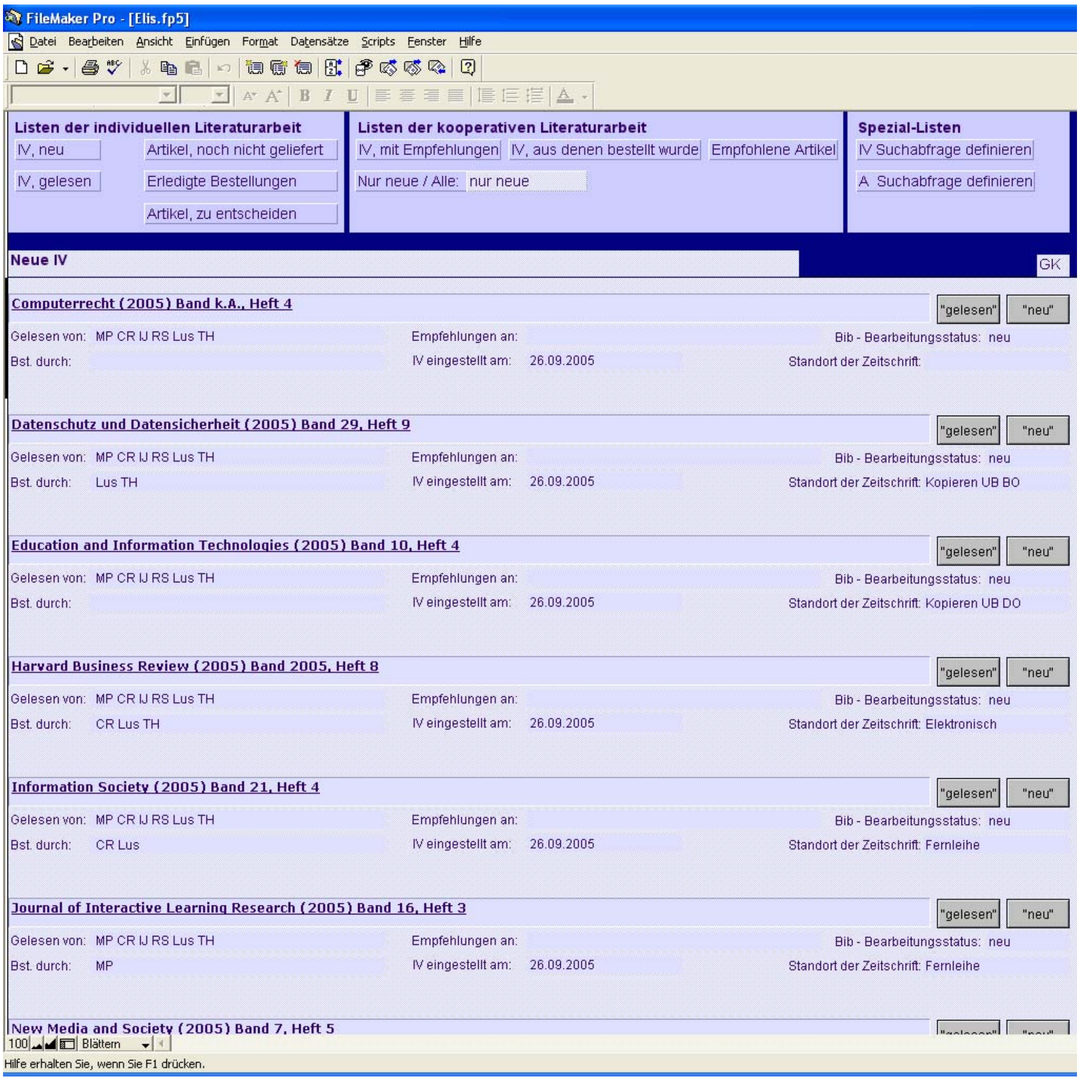
b) References.fp stores information related to single articles.

c) Zeitschriften.fp stores information related to journals.

The workflow logic is included in the GUI of the Filemaker® databases.

For the users there are two sets of GUIs: one for the scientists and another one for the student workers in the library (cf. Figure 56 for a screenshot). The GUIs for the scientists include:

- A personalized list of tables of content that are new for the user (cf. Figure 24 for a screenshot)
- Functions to view lists of articles within one table of contents(cf. Figure 10 for a screenshot)
- Functions to view detailed information for selected articles from a table of contents(cf. Figure 25 for a screenshot)
- Functions for ordering an article (cf. Figure 25 for a screenshot; the functions and information regarding the ordering of articles are located in the top part of the GUI which is headed "Bestellungen")
- Functions for recommending an article to a colleague for reading (cf. Figure 25 for a screenshot; the functions and information regarding recommendations of articles are located in the middle part of the GUI which is headed "Empfehlungen")
- Awareness
    - for the tables of content read by others (cf. Figure 25: the table of content, in which the article selected here, is contained has been read by users MP, AC, CR, TH, Lus, IJ, GK, AK);
    - for the articles recommended to oneself and to others (cf. Figure 25: this article was recommended by user GK to user TH);
    - for articles ordered by others (cf. Figure 25: the article selected here has been ordered by GK);
    - for the status of the library's work on tables of content and individual articles (cf. Figure 24: the tables of content listed here have not yet been processed by the library team – their status "Bib – Bearbeitungsstatus" is "neu"; cf. Figure 25 the table of content, in which the selected article is contained, has already been processed by the library team, its status is "Bst_fertig_bearbeitet").
-


FileMaker Pro - [Elis.fp5]
Datei Bearbeiten Ansicht Einfügen Format Datensätze Scripts Fenster Hilfe
Listen der individuellen Literaturarbeit
IV, neu
IV, gelesen
Artikel, noch nicht geliefert
Erledigte Bestellungen
Artikel, zu entscheiden
Listen der kooperativen Literaturarbeit
IV, mit Empfehlungen
IV, aus denen bestellt wurde
Empfohlene Artikel
Nur neue / Alle: nur neue
Spezial-Listen
IV Suchabfrage definieren
A Suchabfrage definieren
"gelesen"
"neu"
100
Blättern
Hilfe erhalten Sie, wenn Sie F1 drücken.

Neue IV GK

Computerrecht (2005) Band k.A., Heft 4
Gelesen von: MP CR IJ RS Lus TH | Empfehlungen an: | Bib - Bearbeitungsstatus: neu
Bst. durch: | IV eingestellt am: 26.09.2005 | Standort der Zeitschrift:

Datenschutz und Datensicherheit (2005) Band 29, Heft 9
Gelesen von: MP CR IJ RS Lus TH | Empfehlungen an: | Bib - Bearbeitungsstatus: neu
Bst. durch: Lus TH | IV eingestellt am: 26.09.2005 | Standort der Zeitschrift: Kopieren UB BO

Education and Information Technologies (2005) Band 10, Heft 4
Gelesen von: MP CR IJ RS Lus TH | Empfehlungen an: | Bib - Bearbeitungsstatus: neu
Bst. durch: | IV eingestellt am: 26.09.2005 | Standort der Zeitschrift: Kopieren UB DO

Harvard Business Review (2005) Band 2005, Heft 8
Gelesen von: MP CR IJ RS Lus TH | Empfehlungen an: | Bib - Bearbeitungsstatus: neu
Bst. durch: CR Lus TH | IV eingestellt am: 26.09.2005 | Standort der Zeitschrift: Elektronisch

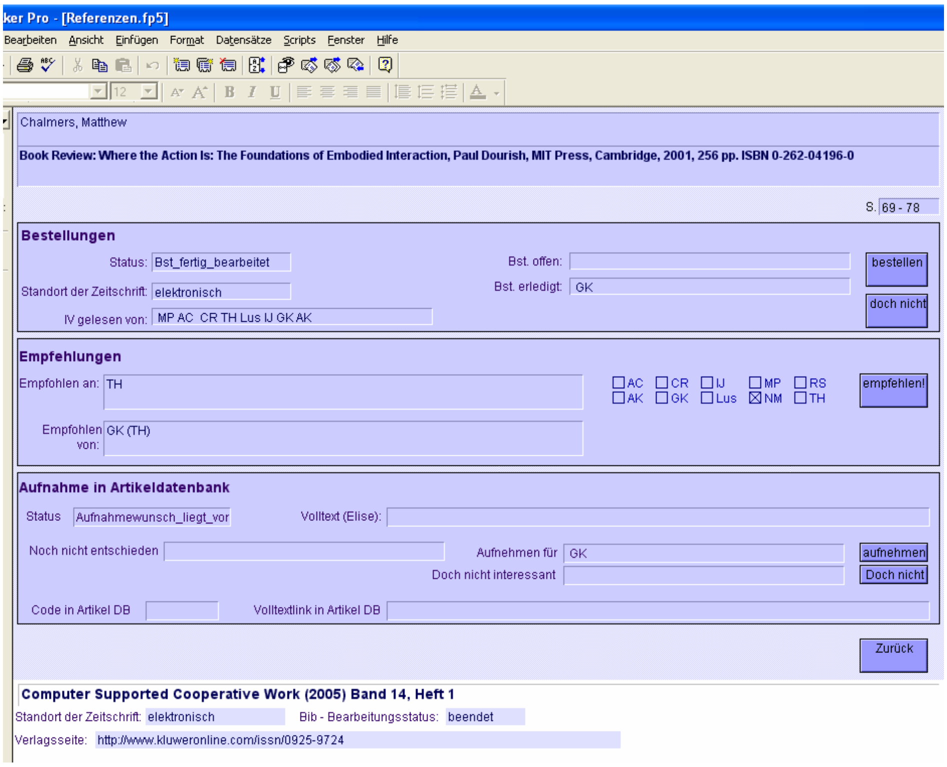
Information Society (2005) Band 21, Heft 4
Gelesen von: MP CR IJ RS Lus TH | Empfehlungen an: | Bib - Bearbeitungsstatus: neu
Bst. durch: CR Lus | IV eingestellt am: 26.09.2005 | Standort der Zeitschrift: Fernleihe

Journal of Interactive Learning Research (2005) Band 16, Heft 3
Gelesen von: MP CR IJ RS Lus TH | Empfehlungen an: | Bib - Bearbeitungsstatus: neu
Bst. durch: MP | IV eingestellt am: 26.09.2005 | Standort der Zeitschrift: Fernleihe

New Media and Society (2005) Band 7, Heft 5

Figure 24: ELISE - List of Tables of Contents which are new for User GK

ELISE provides more functions for collaborative work on scientific literature that support the phase once an article has been delivered and read. At this point the user can decide on the further processing of the article e.g. whether or not it should be included into the team's literature database (the lower part of the GUI in Figure 25, which is headed "Aufnahme in Artikeldatenbank" includes some of these functions). But within this thesis I concentrate on the functions mentioned above, because they are the once that replace the original paper-based procedure.


ker Pro - [Referenzen.fp5]
Bearbeiten Ansicht Einfügen Format Datensätze Scripts Fenster Hilfe
12
bestellen
doch nicht
empfehlen!
aufnehmen
Doch nicht
Zurück

Chalmers, Matthew

Book Review: Where the Action Is: The Foundations of Embodied Interaction, Paul Dourish, MIT Press, Cambridge, 2001, 256 pp. ISBN 0-262-04196-0

S. 69 - 78

Bestellungen

Status: Bst_fertig_bearbeitet

Standort der Zeitschrift: elektronisch

IV gelesen von: MP AC CR TH Lus IJ GK AK

Bst. offen:

Bst. erledigt: GK

Empfehlungen

Empfohlen an: TH

☐ AC ☐ CR ☐ IJ ☐ MP ☐ RS
☐ AK ☐ GK ☐ Lus ☒ NM ☐ TH

Empfohlen von: GK (TH)

Aufnahme in Artikeldatenbank

Status Aufnahmewunsch_liegt_vor

Volltext (Elise):

Noch nicht entschieden

Aufnehmen für GK

Doch nicht interessant

Code in Artikel DB

Volltextlink in Artikel DB

Computer Supported Cooperative Work (2005) Band 14, Heft 1

Standort der Zeitschrift: elektronisch

Bib - Bearbeitungsstatus: beendet

Verlagsseite: http://www.kluweronline.com/issn/0925-9724

Figure 25: ELISE - Detailed Information about an Article

6.3.3 Interventions and Results

The case study „electronic ToC-service" took place within my workgroup, and all tasks were accomplished within the group: design and implementation of ELISE as well as the consultancy towards new work procedures including the facilitation of workshops. Due to the workgroup's moving from one university to another, there was a major interrupt in the project from about summer 2004 to spring 2005. In June 2004 the case study "electronic ToC-service" had reached the same level as the case study "mobile communication system" at the end in April 2004. With the relaunch of the project in April 2005, the phases deployment and use were added.

The interventions relevant for this thesis consisted in facilitated STWT-workshops and the usage of semi-structured diagrams to represent the group's socio-technical self-description. Section 11.1.3 (pp. 301) lists all workshops conducted during the case study and section 11.3.2 (pp. 332) contains a representative set of diagrams from the case study.

The case study "electronic ToC-service" resulted in a socio-technical organization in the sense of Term 20 and in a socio-technical setting in the sense of Term 22. The CSCW-system ELISE which is operational since May 2005 was designed and implemented; and new work procedures, that are described by new socio-technical self-descriptions, were planned and established. Thus both results that are required from a CSCW-project in Figure 11 were achieved in this case study.

6.4 Contributions of the Two Case Studies

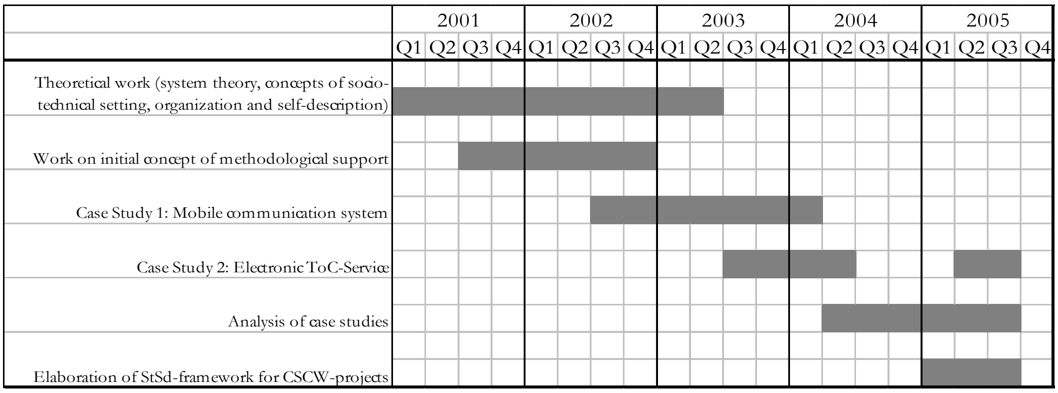
Both case studies contribute to the further development of the initial design of methodological support. However due to their different settings and contents, their contributions to the analysis in chapter 7 differ. This section describes how the case studies compare with each other and what kind of material they contribute for the analysis.

6.4.1 Chronology of Events

The phases in which the framework for using StSd (chapter 8) has been derived did not take place as sequentially as might be suggested by the sequential ordering of chapters needed in a text. Figure 26 illustrates the relative timing of the research phases I describe in this thesis.

The sequence and the overlapping of phases are indicators how they have influenced each other. Case study 2 was behind case study 1 in the early phases (up to the qualification workshop in June 2004) and benefited from the experience. Similarly the work on the theoretical concepts continued after the initial concept for the case studies had been defined. In fact, it is still an ongoing task, but the status reached in 2003 is the basis for this thesis.

The CSCW-system ELISE is operational and case study 2 (electronic ToC-service) is an ongoing project within my workgroup; the part of the project that is relevant for this thesis ended on August 22nd 2005 with an evaluation workshop



	2001				2002				2003				2004				2005			
	Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4
Theoretical work (system theory, concepts of socio-technical setting, organization and self-description)	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨										
Work on initial concept of methodological support			▨	▨	▨	▨	▨	▨												
Case Study 1: Mobile communication system							▨	▨	▨	▨	▨	▨	▨							
Case Study 2: Electronic ToC-Service											▨	▨	▨	▨				▨	▨	
Analysis of case studies														▨	▨	▨	▨	▨	▨	
Elaboration of StSd-framework for CSCW-projects																	▨	▨	▨	

Figure 26: Relative Timing of Research Phases

6.4.2 Material available for Analysis

The following categories of data were collected during the case studies:

- **Diagrams**: all diagrams created during the workshops have been archived in a chronological manner. They are the main source of data for the analysis. For each case study a representative set of diagrams is included in the appendix 11.3 (pp. 308 for case study „mobile communication system" and pp. 332 for case study „electronic ToC-service").

- **Audio- and / or video-recordings**: all STWT-workshops and some of the intermediate meetings were recorded. Some sequences were transcribed; examples of those that I use for argumentation are included in appendix 11.4, pp. 359.
- **Photos** from ethnographical fieldwork in case study 1: in the analysis phase of the project observational interviews were conducted and photos were taken for documentation (cp. appendix 11.1.1, pp. 299 for details).
- **Photos** from artifacts (e.g. pin boards) created during the workshops: when issues were documented on flipcharts or pin boards during workshops, photos were taken such that these artifacts are available for analysis (e.g. Figure 44, Figure 45, or Figure 47).
- **Interviews**: in case study 1, interviews that included questions related to the usage of semi-structured diagrams were conducted with the participants of the workshops. These interviews are available as audio files as well as in form of transcripts. One transcript of an interview used for argumentation within section 7 is contained in appendix 11.5, pp. 365.
- **Questionnaires**: in the final phase of case study 2 questionnaires were sent out to the participants of workshops via e-mail. They are included in the appendix 11.5.2.
- **Statistics of use**: for case study 2 a logbook was kept for 6 weeks documenting the reading and ordering of articles. Table 40 in appendix 11.6 contains a summary.
- **E-Mail archive:** for case study 2, all e-mails exchanged among the participants on the topic of “electronic ToC-service” are archived. They contain feedback as well as organizational requests and regular notifications
- **Participants’ feedback**: some workshops were concluded by a feedback question that related to the techniques employed in the workshop. The answers given by the participants are either included in the recordings of the workshop or separate documents with summaries exist.
- **Researcher’s diary**: since I was directly involved in the projects, I kept a research diary to document experiences and ideas that are not contained in the other material; either because they were personal insight resulting from the personal involvement, or because they came up in informal conversations in-between sessions.

Section 11.1 in the appendix contains an overview of the empirical work and lists the material available for the individual interventions (pp. 299 for case study 1, pp. 301 for case study 2).

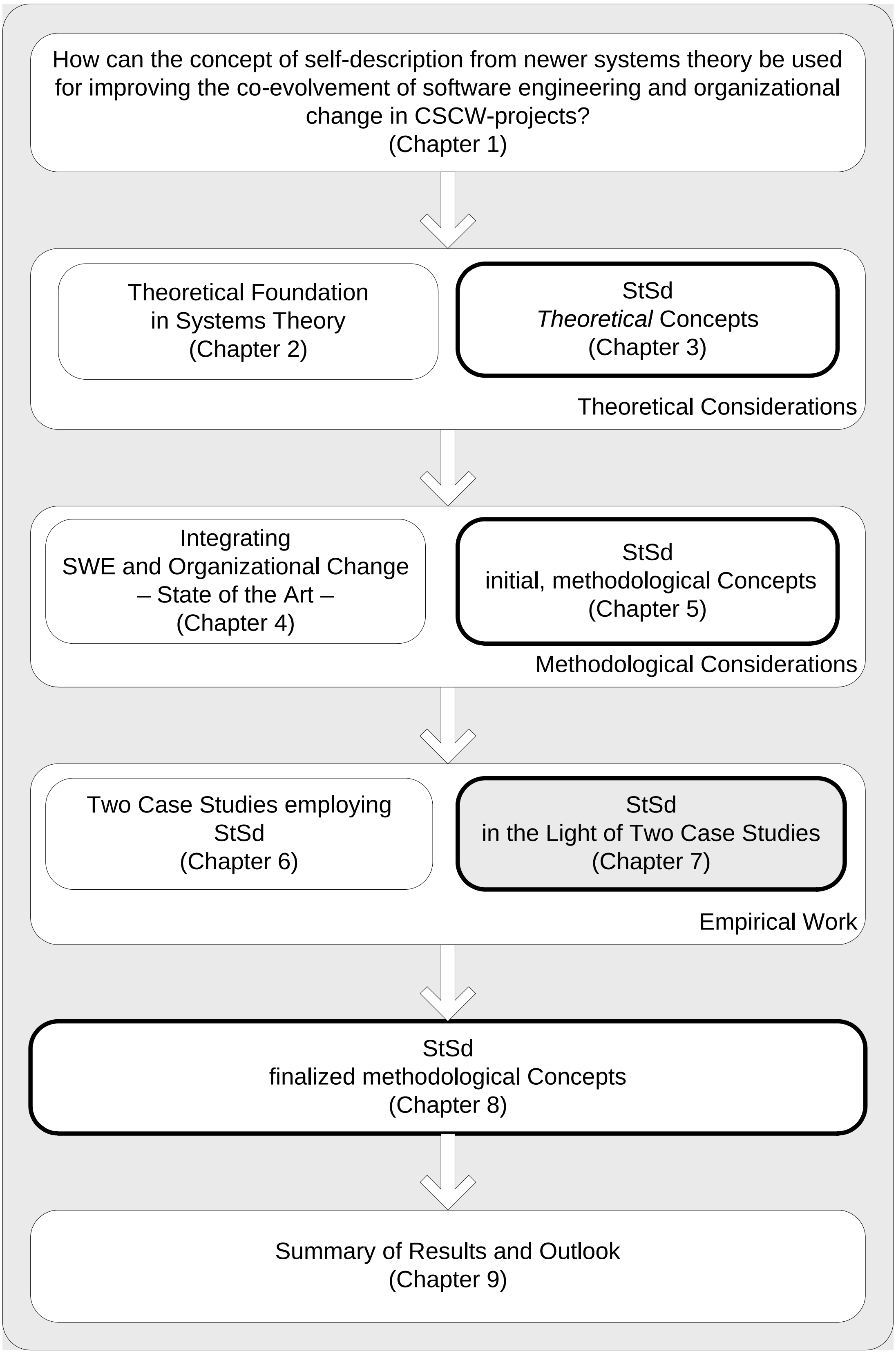
6.4.3 Comparison of Case Studies

While the two case studies are both concerned with the design, implementation and deployment of a CSCW-system, they differ from each other in their setting. These differences contain both advantages and disadvantages with respect to the analysis in chapter 7; these are summarized in Table 17.

	Mobile Communication System	**Electronic ToC Service**
Strength	a) Software-development process with a division of labor between the participating research groups that required coordination; this setting is well suited to evaluate the effects of an additional type of documentation in the project. b) Software-development process with independent groups of stakeholders: users, managers, developers, consultants came from different organizations.	a) The Case Study describes a complete project in the sense that all phases from inception to deployment and usage took place. The analysis of this case study can provide further insights into the effect of socio-technical self-descriptions in the phase of usage. b) Great flexibility in using the diagrams because participants were familiar with modeling notation as well as with the supporting editor.
Potential Weakness	a) Research project – participants knew that this prototype would not be deployed in real. b) Organizational restructuring took place during the project and influenced who participated in the workshops (dispatchers that were outsourced into the project „Westkreis" when the project started, were employees of the steel trading company again at the end of the project).	a) Development and use of the software happen within one organization, so that the need for communication between different stakeholders is reduced in comparison to an industrial project. b) Users belong to my own research group so that participants may be positively biased towards the project.

Table 17: Comparison of Case Studies

The data selected for analysis in section 7 takes the potential strengths and weaknesses of each case study into account; special issues are discussed where appropriate.


How can the concept of self-description from newer systems theory be used for improving the co-evolvement of software engineering and organizational change in CSCW-projects?
(Chapter 1)
Theoretical Foundation in Systems Theory
(Chapter 2)
StSd
Theoretical Concepts
(Chapter 3)
Theoretical Considerations
Integrating SWE and Organizational Change – State of the Art –
(Chapter 4)
StSd
initial, methodological Concepts
(Chapter 5)
Methodological Considerations
Two Case Studies employing StSd
(Chapter 6)
StSd
in the Light of Two Case Studies
(Chapter 7)
Empirical Work
StSd
finalized methodological Concepts
(Chapter 8)
Summary of Results and Outlook
(Chapter 9)

7 StSd – in the Light of Two Case Studies

This chapter is dedicated to the analysis of the case studies. The purpose of the case studies is to employ my initial methodological concepts for StSd described in chapter 5 in form of action research projects, and to supply empirical insights that can be combined with the theoretical considerations to elaborate a framework for the usage of socio-technical self-description in CSCW-projects (chapter 8). The contents and goals of the case studies as well as the research methods are described in section 6.

The analysis follows the research questions that are summarized in Table 14. The following sections contain subsections labeled „**Example**". These contain detailed descriptions of episodes from the case studies; e.g. parts of a diagram with a description of the discussion related to it; transcripts from interviews; photos from workshops etc. The empirical material is described as objectively as possible, without interpretation. The latter follows in the other subsections. Here the empirical material is discussed and interpreted, compared to other pieces of material or to literature (cf. section 6.1, pp. 153 for a discussion of the methods for developing grounded theory that are employed here). A complete list of examples drawn from the case studies is included at pp. 290.

7.1 Content

The first set of questions for the analysis of the case studies is concerned with the content of socio-technical self-description. The initial concept derived in section 5.2.1 made no detailed assumptions about the contents of StSd; therefore the analysis of the case studies starts with an analysis of the contents of the diagrams created.

7.1.1 C1: What are the Topics in StSd?

At the time when the two case studies started, only initial ideas about the contents of socio-technical self-descriptions existed. The overall concept implied that work-procedures must be represented in their relation to the software-system. With this in mind the first diagrams were created; from then on the needs in the projects combined with the goal to support the structural coupling of organization and software-system determined the contents of the diagrams. In order to systematically describe the contents of StSd an ex-post analysis of all diagrams was performed during which I sorted the contents of the diagrams into thematic clusters eminent in both case studies. No claim is made that these clusters are mutually exclusive or that their list is exhaustive.

The thematic clusters do overlap and influence each other; nevertheless I consider it helpful to analytically separate them for discussion.

Thematic Cluster	**Section**
Work-Procedures	7.1.1.1
Relation to Software-Functions	7.1.1.2
Agreements concerning original cooperative task	7.1.1.3
Agreements concerning the usage of the CSCW-system	7.1.1.4
Usage of additional technical systems	7.1.1.5
Meta-level comments concerning the project	7.1.1.6

Table 18: Thematic Clusters in Socio-Technical Self-Descriptions

The first cluster is **work-procedures**: all diagrams contain descriptions of work-procedures; i.e. a sequence of actions carried out by the users of the software to fulfill their work assignment. The second cluster **relation to Software-Functions** contains those parts of the diagrams that describe functions of the software in relation to the work-procedures. While user manuals describe what a software-system can do, StSd describe how factions of the software are incorporated into the work-procedures. This includes the conditioned use of certain functions as well as agreements not to use certain functions at all. The third and fourth clusters contain those parts of the diagrams that result from two kinds of cooperative agreements. First there are agreements that pertain to the original cooperative work; that is the work that should be supported by the software-system. The diagrams of both case studies include **agreements concerning the original cooperative task**. A second type of agreements is needed to organize the usage of the software-system. The results are documented in **agreements concerning the usage of the CSCW system**. The fifth cluster contains documentation about the **usage of additional technical systems** that coexist with the new software-system. The topic here is not the technical integration of legacy system. In most modern organizations the functionalities of IT-systems overlap; it proved to be helpful to explicitly decide which system is to be used for which purpose. The last category is concerned with **meta-level comments concerning the project** which are attached to the notational symbols and contain requests for change or reminders for issues still unsolved.

Table 18 summarizes the thematic clusters and names the sections that discuss the cluster in detail.

7.1.1.1 Work-Procedures

The basis of all diagrams created during the case studies are activities that constitute work-procedures. The fact that the projects started out with diagrams representing work-procedures is not surprising; it has been prepared by practical as well as by theoretical considerations elaborated in section 5.2.1.

In case study „mobile communication" my colleague and I created the initial diagram representing the status quo of the work-procedures as a result of our interviews (cf. Figure 62 in the appendix 11.3.1). In case study „electronic ToC service", a student worker from the library team produced diagrams representing the processes of literature acquisition and usage within the group (Figure 86 – Figure 91 in the appendix).

What is remarkable however is the fact that with one exception, work procedures remained to be the subject of modeling throughout both projects. At least in case study „Electronic ToC service" all participants had the knowledge and the power to include more social context into the diagrams, but nobody did.

Before discussing this observation, the following sections give examples for the kind of work-procedures modeled in the case study as well as some thoughts about their granularity and scope.

Example 1: Initial Diagram in Case Study „Mobile Communication"

In the case study „mobile communication" my colleague and I prepared a first diagram that showed the drivers' and dispatchers' work-procedures as we understood them from their ethnographically oriented analysis (cf. Figure 62 in the appendix 11.3.1). In the workshop STWT-1 we discussed the diagram with the drivers and dispatchers and changed it according to their input. The result was a second version of the diagram which was approved by all participants (cf. Figure 64 in the appendix 11.3.1)

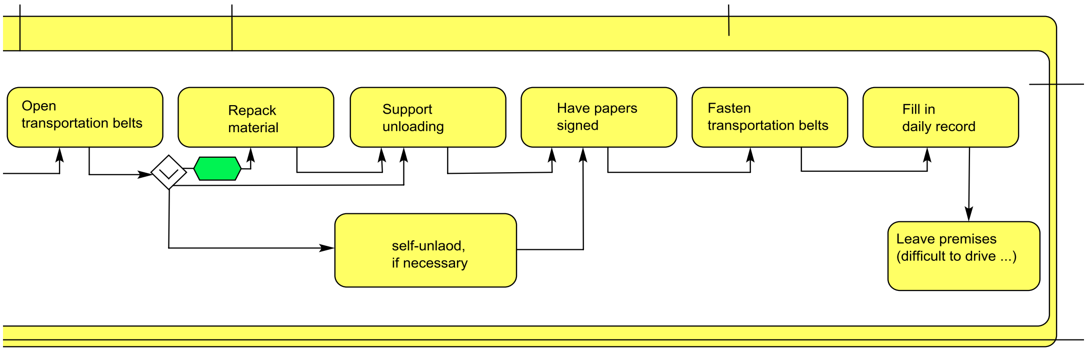

Open transportation belts
Repack material
Support unloading
Have papers signed
Fasten transportation belts
Fill in daily record
self-unlaod, if necessary
Leave premises (difficult to drive ...)

Figure 27: Actions added by the Drivers

The first diagram includes the tasks *support unloading* to indicate that the driver does help to unload the truck although formally he is not obliged to do so. To denote the activity the driver engages in between two customers, the task *driving* (outside the detail of Figure 27) is included.

After the first diagram had been presented in the workshop, the facilitator asked: „Is there anything else? If I gave the diagram to the software-developers now, is there anything you would like to change?" One of the drivers spoke up and said: „Yes … that sounds so perfect up there. The driver does his job, all according to plan, and it is done … „. Then he gave examples of things that could go wrong. They all had to do with unexpected obstacles that could be avoided if the driver had better information about the customer's site (cf. to section 11.4.1 for a transcript). As a result of the discussion two actions were added to the diagram.

First, the action *Self-unload, if necessary* to note that there are situations in which the drivers have to do the unloading alone. Second, *Leave premises (difficult to drive …)* to distinct between driving on a road between two customers and the effort to leave a customer's site which might be difficult due to narrow driveways, muddy roads or other obstacles. Figure 27 which is a detail from Figure 64 in the appendix 11.3.1 shows these tasks.

What should be Included in StSd in early Phases of a CSCW-project?

The theme of the first diagram was to represent the current work-procedures in the „Westkreis". The phrase „current work-procedures" allows a broad range of interpretation and it is as difficult as important to decide what should be included at what level of detail. The two sides that need to be balanced are: the provision of much and detailed information with the intention to create a rich and meaningful diagram on the one side; and the need to concentrate on the essential aspects in order to reduce complexity and thereby create manageable diagrams on the other.

In the example the drivers requested to add the two tasks *self-unload, if necessary* and *Leave premises (difficult to drive …)*. Most likely, a mobile communication device would not help with either of them; and there would be no interaction with the software. The tasks are not even related to organizing the cooperative work or the usage of the software; so why should they be included in the diagrams? One reason is simply that they were important to the drivers' self-description of their work. They spoke up and suggested a change in the diagram to which all other participants agreed. The second reason is that there is indeed a thematic connection to the mobile communication system: both activities are additional burdens that could be avoided if the drivers had up-to-date information about the customers' premises. The inclusion of the two tasks illustrates the advantages of up-to-date information for the drivers' work-procedures. To include such actions in diagrams of early phases supports the mutual understanding for the work-procedures and their inherent problems. This is especially true for distributed work situations in which the co-workers do not witness the others' problems during their work.

In the early phases of the project the diagrams resemble Checkland's rich pictures (cf. section 4.3.3, p. 118) in so far that everything that supports the discussion between the stakeholders should be included into the diagrams. During the project the diagrams become narrower in the sense that they concentrate on those parts of the work-procedures that bear some (indirect) relation to the CSCW-system. The diagrams become wider at the same time as they include additional actions and agreements related to the CSCW-system.

In both case studies the early diagrams contained actions that were dropped in the course of the project because it turned out that they bore not significant relation to the software-system. *Open transportation belts* and *Repack material* are activities from diagram Figure 62 (Appendix 11.3.1) that vanished from the diagrams.

Other actions that were suspected to be irrelevant in the beginning turned out to be of importance and remained in the diagrams until the very end of the project as the following Example 2 illustrates.

Example 2: Tasks not related to the CSCW-System

Example 1 relates to a diagram in the early phase of the case study „Mobile Communication". The second example takes one of those tasks from the first diagram that seems to have no connection to the software to be developed, and traces its appearance in later diagrams: The task *Fasten transportation belts* from Figure 27. Originally it was included into the diagram because it was a time-consuming task that they felt should not be ignored in the description of the status quo.

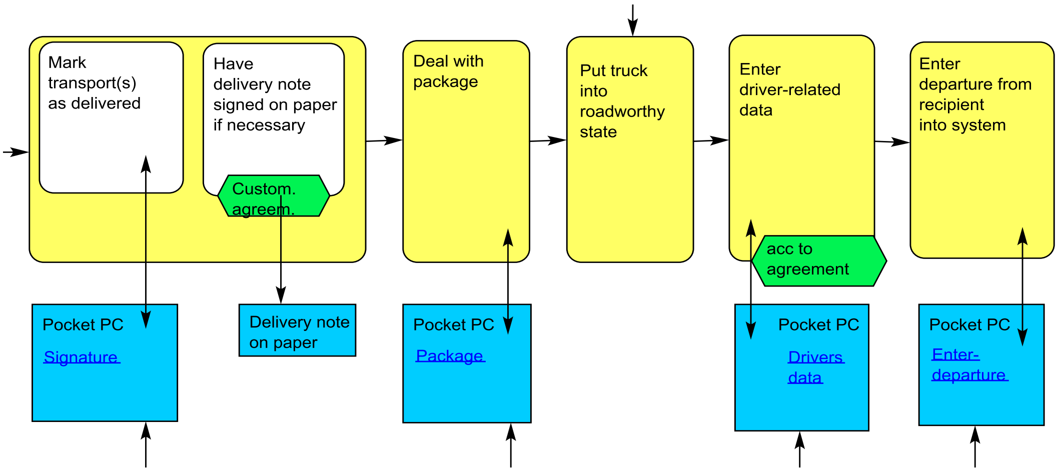

Mark transport(s) as delivered
Have delivery note signed on paper if necessary
Custom. agreem.
Deal with package
Put truck into roadworthy state
Enter driver-related data
acc to agreement
Enter departure from recipient into system
Pocket PC
Signature
Delivery note on paper
Pocket PC
Package
Pocket PC
Drivers data
Pocket PC
Enter-departure

Figure 28: Actions Providing Context for Usage of Software

Figure 28 shows a detail of Figure 78 (appendix 11.3.1) which was used during the validation of the GUI-screenshots and that still contains the action *Put truck into roadworthy state*, which is a more generic form of the task *Fasten transportation belts again* from the first diagram (Figure 62 in the appendix). It seems surprising that this task still exists in diagrams that are used for the validation of GUI-screenshots, but it serves an important purpose: Its inclusion provides context information for the task *Enter departure from recipient into system* by which the driver informs the dispatcher that the delivery for the customer has now been completed. During the STWT-workshops it was repeatedly discussed whether the information that the delivery note has been signed by the recipient is sufficient for the dispatcher or not (cp. also diagram in Figure 67 in the appendix where this topic is noted in form of a comment). The participants finally agreed that an additional notification is needed when the driver actually leaves the customer's site. This notification is given with the action *Enter departure from recipient into system* and it only makes sense if it is understood that there are time-consuming tasks to be done after the delivery note has been signed. The socio-technical diagram shown in Figure 28 and Figure 78 (in the appendix) serves this purpose.

Example 3: Modeling Social Context other than Work-Procedures

As mentioned above there is only one example in both case studies that made use of additional symbols to model social context exceeding work procedures. In case study „Mobile Communication" the following problem became apparent: the drivers complained that their papers (i.e. delivery notes) contain out-of-date information about the recipients on their tour; and that this causes delays as well as additional unnecessary efforts.

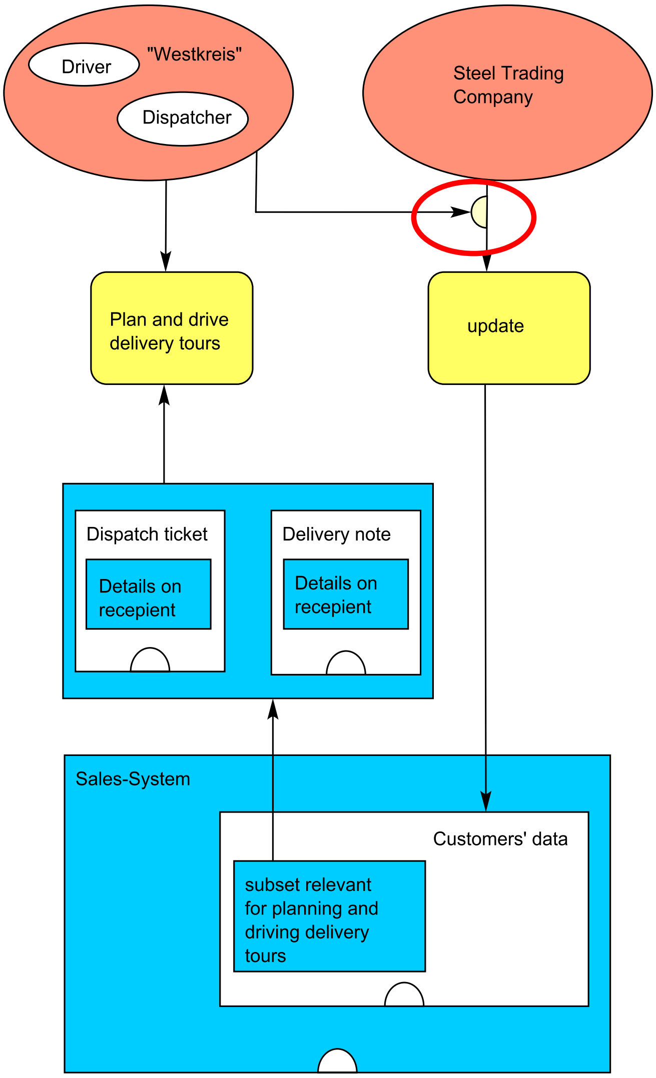


Figure 29: Example of Modeling Social Context

This problem was not new but had existed from the very beginning of the project; it is caused by the fact that „Westkreis" is an outsourcing project and by the legacy system *sales system* used by the *steel trading company* to handle their customers and sales. The dispatchers from the „*Westkreis*" have read-only access to the data and any request to correct a customer's dataset needs to be passed from the driver to the dispatcher and on to the sales person in the *Steel Trading Company*. Figure 29 contains the diagram I prepared to discuss the issue with the manager of „Westkreis". It includes a relation from *„Westkreis"* to the relation between *Steel Trading Company* and *update* to indicate that *„Westkreis"* has an interest in the activity performed by the *Steel Trading Company*.

Modeling Social Context in StSd

SeeMe provides numerous constructs to model social context [Herrmann et al. (2004a)]. The one employed in Figure 29 is „has an interest in"; it shows very well that the roles *Driver* and *Dispatcher* that use the data *details on recipient* do not carry out the action *update*.

The arrow from the role *Westkreis* to the arrow leading to the activity *update* indicates that the *Westkreis* is interested in the execution of the process *update* but that it cannot execute it itself. In both case studies there are numerous examples that could have been modeled in this way. Literature also suggests additional diagrams supplementing business processes with social context (e.g. goal-exception- and dependency-diagrams by [Katzenstein, Lerch (2000)]). Why were such constructs not used during the case studies?

The assumption is that they are useful on a meta-level of organizational development. In a situation where an organization is searching for causes of problematic situations it can be helpful to model combinations of interests or dependencies. Such diagrams can then be used to uncover conflicts of interests; but their contribution is more of explanatory and less of guiding nature. StSd however, should serve as guidance for the organization's members (cf. Term 21) and for this modeling the people's activities is more helpful than modeling their interests.

To stay with the example of the problem described above, an StSd would include an agreement between the participants how requests for corrections are passed from the *driver* to the *Steel Trading Company* and how fast the latter promises to update the database. The discussion and solution to this problem was outside the scope of the case study, because the Steel Trading Company had no part in it. Nevertheless, Figure 85 in the appendix sketches a socio-technical solution as discussed in the project: The prototype of the mobile communication system allows the driver to locally update the recipient's data on his mobile device and transfer this update as a suggestion to the main database. The dispatcher is responsible for keeping the data and checking the suggestions made by the drivers. As all the others this diagram relies on roles and activities and does not make use of any additional constructs to represent the social context.

7.1.1.2 Relation between work-related Activities and Software-Functions

The next category regarding the content of socio-technical self-description is the „relation to software-functions". Once the work-procedures have been described, the question is how they are or will be supported by a CSCW-application.

Example 4: Single Actions Augmented by Links to Software-Functions

As elaborated in chapter 3 and section 5.1 socio-technical self-descriptions need to describe the structural coupling between a social system and the new software-system in its environment. One way to do this is to document how software-functions are used during the work-processes. To this end, each step in the work-procedures is augmented by the software-functions that are intended for its support. Figure 30 (which is a detail of Figure 78 in the appendix) gives an example from the case study „Mobile Communication". Most of the actions are connected to entities marked *Pocket PC* which refer to the driver's hand-held device. Within each entity *Pocket PC* a reference to the specific software-function that is meant to be used during the activity is given by a hyperlink. Each hyperlink refers to one or more screenshots representing the workflow within SpiW-Com.

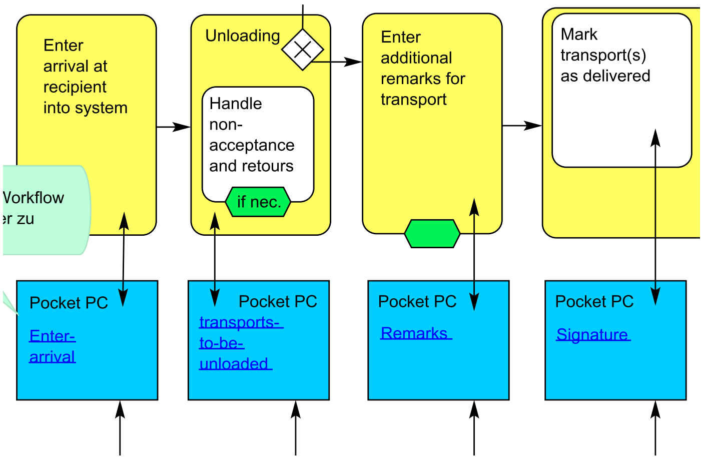

Enter arrival at recipient into system
Workflow
er zu
Unloading
Handle non-acceptance and retours
if nec.
Enter additional remarks for transport
Mark transport(s) as delivered
Pocket PC
Enter-arrival
Pocket PC
transports-to-be-unloaded
Pocket PC
Remarks
Pocket PC
Signature

Figure 30: One-to-One Relation between Actions and Software-Functions

Example 5: Clusters of Actions Augmented by Links to Software-Functions

While Example 4 shows how single steps within work-procedures are connected to single software-functions, Figure 31 (detail from Figure 76 in the appendix 11.3.1) shows an example from the same case study where a vague cluster of activities is connected to a list of software functions: the driver is on the road and approaches the customer's site; he now has to find the exact place to unload the truck and the person responsible for receiving the goods. Especially in the case of a new customer but also in case of large industrial areas this can be a difficult and time-consuming task. Workflows with supporting software-functions are not a suitable means to represent such situations. The activities shown (*Locate recipient*, *Approach site*, *Find contact person*, *Drive to loading point*) are meant to give an impression of the situation the driver is in. The fact that the activity *find contact person* is included shows that this is actually a task that may pose a problem and is therefore worth noting. Furthermore the activities listed indicate the kind of information needed by the driver in such a situation. For the software the diagram shows which menus could be used by the driver to look for information (*AST*, *Instructions*, *Info-Customer*, *Info-Loading-Point*).

How detailed should References to Software-Functions be in StSd?

Figure 30 and Figure 31 provide quite different examples for the level of detail on which references to the software-functions can be made. On the one side the activities are described with regard to their task-related aspects only and the handling of the CSCW-system is merely implied by the relation between the activity and the entity *Pocket-PC*. The tasks *Unloading* in Figure 30 and all of the tasks represented in Figure 31 are examples for this. On the other side actions describing the handling of the CSCW-system can be explicitly included as the action *Enter arrival at recipient into system* in Figure 30 does. Sequence diagrams, that are part of UML's interaction view [Rumbaugh, Jacobson, Booch (2005), p. 27, 38] and that model each interaction between parts of the system, are on the far end of this side.

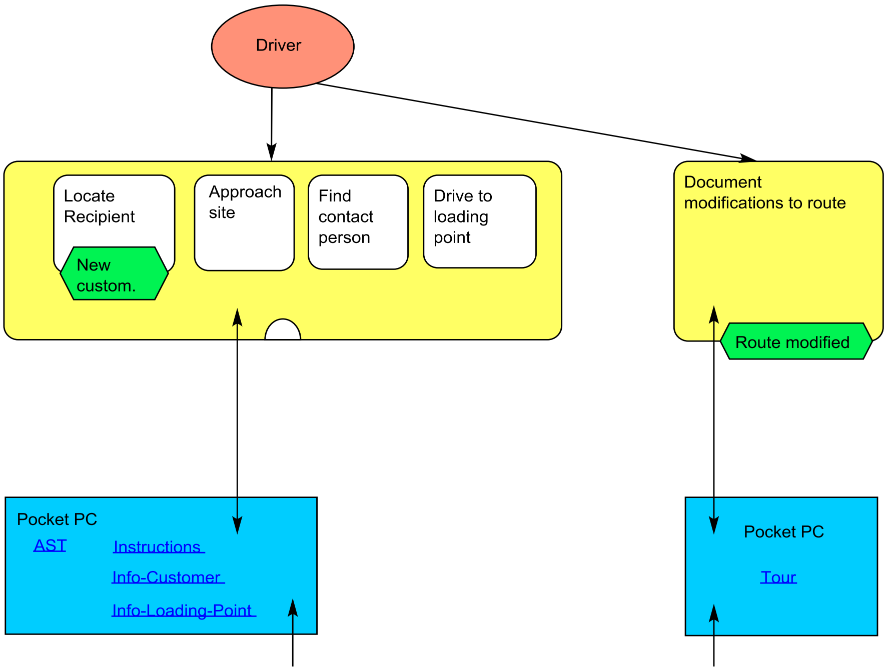


Figure 31: Rough Cluster of Actions from Case Study „Mobile Communication“

The granularity of the links to software-functions should aim to match that of the work-procedures. The diagrams should document which actions will make use of the software-system and which will not. Detailed pictures such as the mentioned sequence diagrams containing each interaction between user and GUI would distract from the context of the work-procedures and move the focus to usability issues which are better discussed with a prototype. Another problem with a very detailed representation of software-functions is that of redundancy. If for example all options for different paths through a GUI are modeled within the StSd, then information that is most likely already contained in the screenshots (or prototype) are duplicated in the diagrams. Which makes a combination of both difficult (cp. also section 7.2.2). It is suggested providing details of the man-machine interaction where needed at a hierarchically deeper level. In case study „mobile communication" GUI-screenshots that showed every possible path through the masks of the GUI documented the interaction between user and software in early phases of the project. The files of the screenshots were connected to the entities representing the pocket PC in the diagrams by hyperlinks; i.e. they could be displayed when more details about the software-functions were needed.

So to ensure that the diagrams can actually provide additional context information for the usage of the software, they should refer to actions of the work-procedures rather than to interactions with the software. Sometimes however it seems necessary to include actions that are interactions with the software. *Enter arrival at recipient into system* in Figure 30 is one example. This particular action is included in the diagram because it is new and it provides information that the dispatchers did not have without SpiW-Com. Some discussions are revolved around this activity, such as when exactly the driver should announce his arrival. It is quite impossible to give an exhaustive list with situations in which interactions with the software should be modeled in greater detail. The following list summarizes situations in which interactions were explicitly modeled during the case studies:

1. When there is no action in the genuine work-procedure that relates to the interaction with the CSCW-system. This is usually the case if information is to be entered that serves somebody else's purposes. *Enter arrival at recipient into system* in Figure 30 is one example because here, the dispatchers benefit from the information; their options to plan the tours are improved if they have access to the information. For the work of the driver it quite unnecessary to enter the data. *Enter driver related data* in Figure 33 is another example. The information entered here is necessary for legal and administrative purposes but not for the driver's work-procedure.
2. When the task consists of interacting with the CSCW-system. The activities *selber eintragen* (*enter oneself*) and *eintragen* (*enter*) in Figure 91(in the appendix) are examples. Here the users have an interest that their literature is included into the database; hence the data need to be entered. This task can either be done by the scientific staff or by the student workers in the library. Another type of activities where the interaction with the CSCW-system is the actual task to be performed are all kind of maintenance actions. *Pool pflegen* (*maintain pool*) with the subtasks *ZS-Stammdaten aktualisieren* (*keep basic data for journals up to date*) and *Benutzerliste aktualisieren* (*keep user data up to date*) in Figure 99 (in the appendix) are examples of this kind.

The examples given so far showed how software-functions supported certain actions in the work-procedures. It turned out that it was equally important to document those features of the software that should not be used or that were subject to certain conditions.

Example 6: Software-Functions that are excluded from Use

The software-system SpiW-Com provides functions to generate item lists for the loading of trucks. In the „Westkreis" such lists are generated using the Steel Trader's software, therefore the corresponding functions in SpiW-Com should not be used. Figure 32 shows how this agreement is documented by the modifier on the relation between the task *load for tour* and the entity *Pocket PC*.

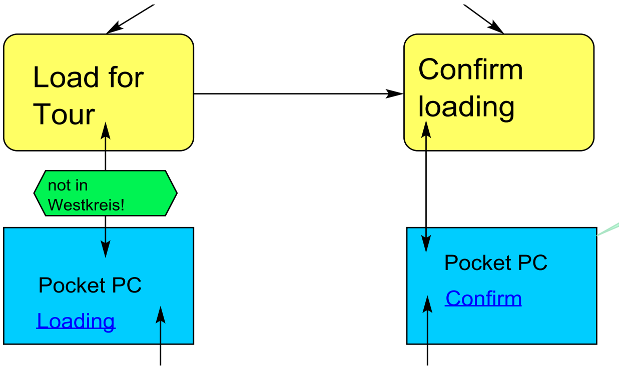


Figure 32: Software-Function that will not be used in the 'Westkreis'

Example 7: New Task that is still subject to debate

In the previous example the task *load for Tour* is a task that has always existed and that is absolutely necessary for the work-procedure. Figure 33 on the other hand shows the task *enter driver-related data* that is new and only exists because it is made possible by the new software. The mobile communication system SpiW-Com includes functions to collect additional data about the drivers' work: begin or work, time on the road, breaks, etc. Some of this data needs to be entered manually by the driver. Since there already were other systems to collect the data needed for legal reasons, the participants in the workshops were unsure whether the additional information that could be gained was worth the effort by the driver. Since the question could not be resolved during the workshops the modifier *subject to agreement* is attached to the task *enter driver-related data.*

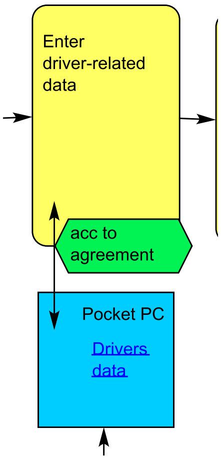


Figure 33: Task that is still subject to debate

Example 8: Need for Organizational Decisions

The last example in this category shows a case where the organization still needs to take a decision between two options that the software offers. There was no dispute over the fact that the mobile client needs a login-function by which the user authorizes himself. A big question however was whether the login should relate to the truck, the driver or both. There are quite different organizational patterns behind each solution:

a) A Pocket PC could be firmly associated with a truck; each driver driving the truck uses the same Pocket PC and has to authorize himself by typing in his name and password.

b) A pocket PC could be firmly associated with a driver; each driver takes the device with him regardless of the truck he drives. In this case he needs to type in the identification of his current truck, because the software-system associates tours with trucks.

c) No association is assumed and the software requires the input of both identifications truck and driver.

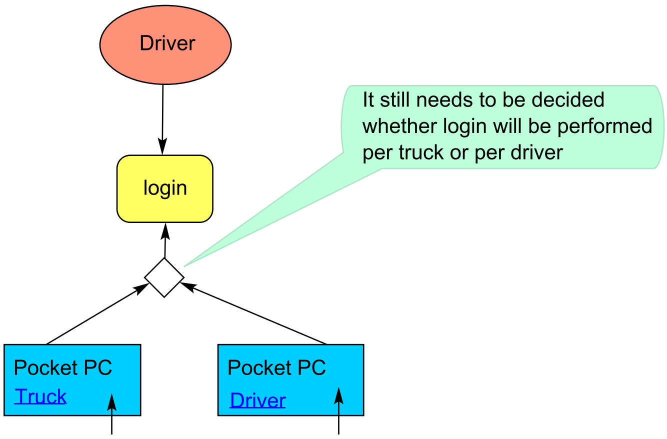

Driver
login
It still needs to be decided whether login will be performed per truck or per driver
Pocket PC
Truck
Pocket PC
Driver

Figure 34: Organizational Option that needs to be decided

When the issue of logging in first came up, the participants could not easily make a decision because of the broad organizational implications. The participating software engineer decided to create the GUI for both options. Figure 34 is a predecessor of Figure 74 (in the appendix). The empty connector in the relation between *login* and *Pocket PC* with link *Truck* and *Pocket PC* with link *Driver* shows that in the software offers both ways of authorization but that the organization has not yet decided which one to use.

First steps towards structural coupling

The previous sections give quite different examples of situations in which the socio-technical diagrams document that certain software-functions will not be used at all or under certain conditions only. The documentation of such conditions is an important task of StSd, because the agreements (including the necessity for further discussion to reach an agreement) are part of the context of use for the software and not of the software itself. Speaking in terms of the categories in Figure 16 they belong to the *formally documented agreements*. The concept of using StSd foresees that these agreements are documented as they are made during the development process, and that they are always presented together with the software. The first advantage is obviously that the agreements are not forgotten and that they do not have to be discussed over and over again. Another important aspect is that the first steps of structural coupling are taken here: The organization agrees on rules how to embed the CSCW-system into its work-procedures.

So the examples above give evidence that structural coupling can start during the development phase of the software.

7.1.1.3 Agreements Concerning the Primary Task

The third thematic cluster of socio-technical self-descriptions is concerned with organizing the cooperative work-procedures. Since CSCW-Systems are intended to support and change cooperative work-procedures, the way cooperation should function once the new system is in place, needs to be discussed and planned in parallel with the design of the technical system itself.

Example 9: Organizing Cooperation

The diagram in Figure 35 is a translation of Figure 69 (in the appendix) and was one result of STWT-2 in case study 'Mobile Communication System'. It shows the cooperation between driver and dispatcher planning a tour

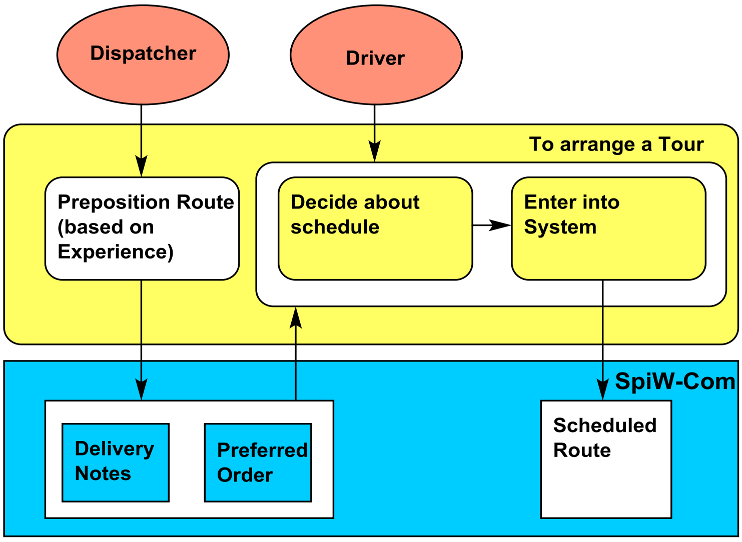


Figure 35: Sorting a Tour into a Route

In the work processes without electronic communication support, the dispatchers planned daily tours for the drivers by arranging the purchase orders to be delivered in piles; one pile for each truck. Although it is the driver's responsibility to sort his tour, the papers were put into an order which suggested a route for the driver. The evening before the tour, the driver leafed through the pile, negotiated some parts of the tour with the dispatcher and rearranged the pile to his liking. This was quite an informal procedure with personal communication during which the dispatcher also explained his reasons for suggesting a certain order. The planning of the mobile communication system initiated a discussion whether and how the easy and informal way of negotiating a tour between driver and dispatcher could be transferred to the new system. The result is shown in Figure 35: The dispatcher should be able to put delivery notes into the system and suggest an order in which the goods should be dispatched. The driver reads this information and rearranges the order of delivery if he wishes to do so. Only after this step has taken place, the system provides a so called „route" for further processing.

Organizational Agreements beyond those inscribed in the CSCW-System

The previous example shows how StSd can capture agreements about the cooperation between actors that go beyond the actual implementation in the CSCW-system. Figure 35 was created during the requirement elicitation phase. In the final implementation

SpiW-COM provided no specific function for the driver to explicitly declare his consent to a route. Since the driver needed a function to rearrange the route in case of unforeseen events anyway, the same function is available for rearranging a tour freshly transferred by the dispatcher. Whether or not the driver makes use of this option in the event of a new tour is up to him and not controlled by the software.

On the technical level the discussion was that the mobile client needed a function that enabled the driver to rearrange the jobs within a tour. On the organizational level the discussion was whether or not the driver was still the one responsible for sorting the tour. The inclusion of the activity *decide about schedule* clearly documents that the new CSCW-system will not imply a change of responsibilities here. There were discussions that the dispatchers should be supported by an additional system for route planning. In this case the order of the tour transferred by the dispatcher would have a completely different meaning, although the functions of the mobile client would not change.

This example shows how the functionality of one CSCW-system can be used for very different ways of organizing cooperative work. StSd offer a place for discussing, deciding and documenting organizational agreements that go beyond those explicitly inscribed in the CSCW-system.

Another organizational topic discussed was how the driver would get the information about tours that was passed on by personal communication before. Example 10 shows how the issue of planning a new tour was continued in the course of the project.

Example 10: Documenting new ways of Interaction

Figure 36 is a detail of Figure 79 (in the appendix) which is a successor of Figure 35 and was part of the training documentation of the drivers. *Sort tour into route* documents that the driver is still the one deciding about the order in which he delivers the jobs. *Retrieve more information* documents that he may need additional information about the tour and that there are two ways to get them: *Retrieve information from system* and *Confer with dispatch*. The latter bears a modifier saying *as little as possible* documenting the intention that the mobile communication system is to be tried first.

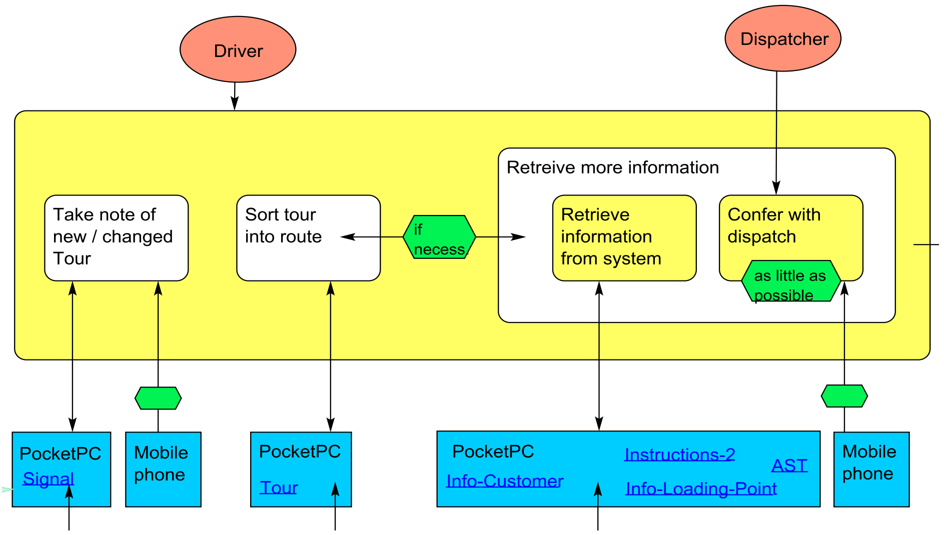


Figure 36: Cooperation when a new Tour is transferred to the mobile client

Example 11: Supporting Activities

Figure 37 is a detail of Figure 94 (in the appendix) created during case study 2; it shows additional organizational tasks assigned to the student workers in the library: they are responsible to regularly remind the scientists to check into the ToC-system, skim through the new ToCs and place their orders. The student workers should do this on a regular basis e.g. using automatically generated e-mails but also by addressing the scientists personally if required.

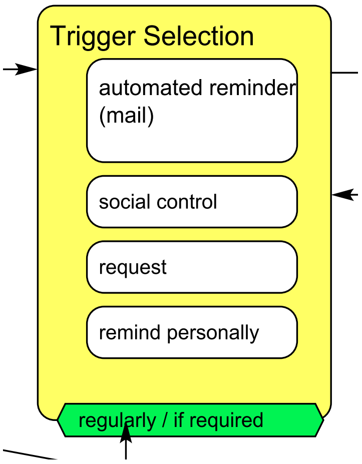


Figure 37: Reminding Scientists to Check ToC-Service

Additional Agreements and Tasks to make the CSCW-System successful

Example 10 from case study 1 and Example 11 from case study 2 are related, because they both show that in order to fulfill the goals connected to the deployment of a CSCW-system, additional agreements are necessary. In case study 1, the drivers had to understand and accept the fact that their phone calls interrupt the dispatchers' work and should therefore be reduced. The diagram in Figure 36 includes the modifier *as little as possible* on the activity *Confer with dispatch*; this diagrams documents that the participants agreed on trying to find information in SpiW-Com first, before they would directly contact the dispatcher.

The electronic Toc-Service in case study 2 was intended to shorten the period of time needed for the ToC to be circled among the scientists. The new CSCW-system brings with it features that serve this purpose. The most important one is that the sequential way of passing the ToC files from one scientist's desk to the next is substituted by a database that allows parallel access by all scientists. While this is an important step, it was clear from the beginning that it will not suffice to speed the process. A scientist who endures a pile of 10 or more files on his or her desk will also endure a long list of unread ToC in the database. So in addition to the new technical support, further measures had to be taken to reach the goal of an accelerated literature work within the group. This is an example how the view on the socio-technical setting as a whole allowed discussing and documenting organizational measures that will add to the success of the technical system. It is clearly documented that the new technical system alone will not be able to solve the primary problem.

7.1.1.4 Agreements Concerning the Usage of the CSCW-System

The previous section was concerned with organizing the cooperative task the users must fulfill. In this section another thematic cluster of socio-technical self-descriptions is discussed: agreements concerning the usage of the CSCW-System. It became evident in both case studies that a software-system even though it was designed employing a participatory approach leaves many option for its usage. At least some of these options could be discussed and planned in advance.

Example 12: Should the Arrival at a Recipient's site always be registered?

Within the qualification workshop of case study 2 the participants discussed whether the action *register arrival* should be performed each time the driver arrives at a recipient's site, even if the truck cannot be unloaded and the driver has to come back later again. This can happen if there is a long queue at the unloading point or if the person responsible for receiving the goods is not available. There was some discussion about the pro's and con's. Finally the participants agreed that the driver should always *register arrival* and then indicate any problems that have occurred. The rationale behind this decision was that the dispatchers need to know about unsuccessful attempts of delivery. Figure 38 documents this decision using a modifier *always upon arrival at recipient's site* on the activity.

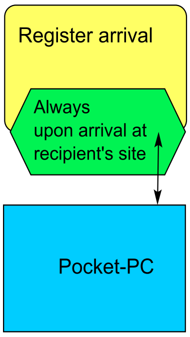


Figure 38: Agreement concerning the Usage of SpiW-Com

Example 13: Agreement on a day for the regular import of new ToC

Figure 39 (which is a translated detail of Figure 105 in the appendix) illustrates an agreement made by the participants of the qualification workshop in case study 2 „electronic ToC-service". The wanted to know how often new ToC would be imported into the system to build up the pool of articles. For their work it is relevant to know whether new ToC would be imported directly upon availability or at other times. All participants agreed that the schedule for import should be transparent. The result of the discussion is shown in Figure 39: the *Library team* is responsible to *trigger import* each Monday.

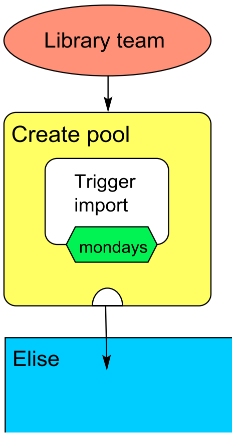


Figure 39: Agreement on a day for the import of new ToC

"Groupware-Conventions" or "CSCW-Systems need additional Agreements"

Example 12 and Example 13 are only two selected from many others that are available in appendix 11.3 which include:

1) Case Study „mobile communication"

   a) The agreement that the drivers register their departure from the recipient's site after they have repacked their truck (cp. Figure 28 and Figure 78 in the appendix).

   b) The agreement that the drivers document any spontaneous change of the route once they arrive at a recipient's site (cp. Figure 66 in the appendix). The alternative would have been to do so immediately when the driver decides to change the route e.g. due to traffic conditions; the advantage would be that the dispatchers receives the information earlier, the disadvantage would be that the driver would have to find a parking place to stop because he may not handle the mobile device while driving.

   c) The agreement that the driver documents a customer's refusal to accept a delivery using SpiW-Com unless the customer insists on an immediate reaction (cp. Figure 82 in the appendix).

2) Case Study „electronic ToC-service"

   a) The agreement that the members of the scientific staff make their selection from new ToC within one week (cp. Figure 105 in the appendix).

   b) The agreement that the library team is only responsible to check ToC with the status „new" for orders place by the scientific staff. Once the status has been moved to „working" or „finished", the scientists must send an additional mail to make their request known (cp. Figure 103 in the appendix).

The need for additional agreements about the usage of the CSCW-systems in both case studies was immense. It is important to note that this need was not caused by badly designed or implemented software. In both cases there were lists with requests for change concerning the software, but the issues collected in these lists did not cause the need for further agreements. The need for agreements concerning the usage of the software is the empirical evidence of the theoretical reflection presented in Figure 16 in section 5.1.2: formal rules crystallized in software invoke the need for further organizational rules to integrate them into work-procedures.

The identification of issues that need to be discussed and the process of finding solutions are parts of the structural coupling that an organization has to undertake if it wants to successfully integrate a CSCW-system into its work-procedures. Software engineers are needed to take part in this process, but they are not the ones to identify, much less to solve the issues. During the discussion that lead to result shown in Figure 38 the software engineer helped to judge the implications of the decision. But speaking in terms of systemic intervention, the responsibility for solving the problem was left with the organization.

The examples do not only show the need for agreements concerning the usage of a CSCW-system, they also show the possibility of doing so. The fact that additional agreements concerning the usage of a CSCW-system are necessary and that they may take the form of groupware-conventions has been discussed before, e.g. by Mark with respect to the POLITeam project (cf. section 3.1.2, p. 68); both case studies confirm the need for conventions. But my findings exceed the findings presented in literature by showing that it is indeed possible to identify open issues in advance and to come to agreements before the CSCW-system is put to use. This is not to say that every open issue will or can be identified in advance, but the experience from the case studies show that a significant number can be identified and taken care of. A side effect of this is that the participants become aware of the fact that there are problems to be solved. It is not (yet) common understanding in CSCW-projects that the integration of a CSCW-system into work-procedures is an additional task to be solved by the organization. Workshops in which such issues are discussed demonstrate to the users the existence but also the resolvability of the task.

7.1.1.5 Usage of Additional Technical Systems

The last thematic category identified in the socio-technical self-descriptions of the two case-studies is the „usage of additional technical systems".

Example 14: SpiW-Com vs. Mobile Phone

In the case-study „Mobile Communication" the usage of mobile phones was one of the biggest issues among the users. From a technical point of view the integration of legacy systems such as the steel trader's booking system or the dispatching software seemed to be most important. For the organization of the daily work-procedures however it was equally important to reach an agreement about the usage of mobile phones

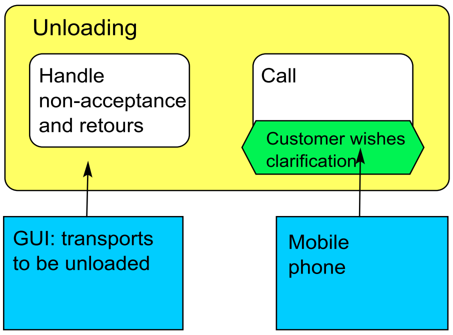

Unloading
Handle non-acceptance and retours
Call
Customer wishes clarification
GUI: transports to be unloaded
Mobile phone

Figure 40: Rule (1) about the Usage of Mobile Phones

During the fourth STWT in the case study 'Mobile Communication System' the participants discussed for every step in the driver's work-procedure whether handy or mobile phone should be used. Figure 40 is one example stating that the driver should use the masks *transports to be unloaded* of the mobile communication system to indicate any problems with the material. If however the customer wants to immediate clarification, the driver should call the dispatcher by phone.

Two other examples that document the usage of mobile phones can be found in Figure 41 (the example on the left is a detail of Figure 78 in the appendix, the one on the right is taken from Figure 79 in the appendix)

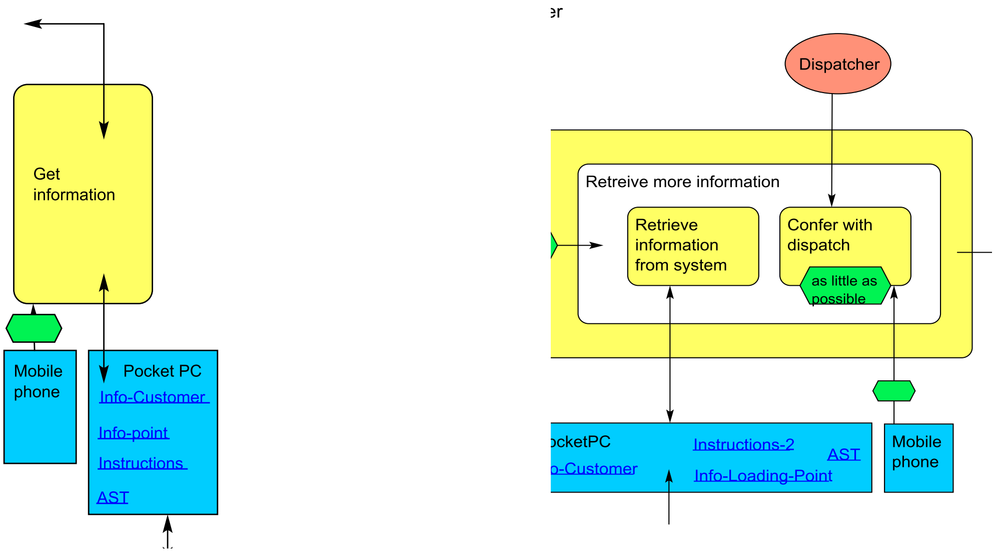


Figure 41: Other Rules concerning the Usage of Mobile Phones

In both examples it is the driver's choice whether or not he uses the mobile phone (indicated by the empty modifier on the relation between *Mobile phone* and the activity). The example on the right includes a second modifier *as little as possible* on the activity *Confer with dispatch* to indicate that it is the wish to reduce the amount of direct communication between drivers and dispatchers.

Example 15: SpiW-Com vs. Paper

In the case study 'Mobile Communication System' paper was another „technical device" that had to be integrated into the new work-procedures. One goal of the project was to reduce paper work. While this could be realized within the „Westkreis", the interface to the recipients of the goods was difficult to influence. Early diagrams contained modifiers for the task of signing the delivery note, indicating that the way the signature is to be obtained is still under debate (cp. Figure 73 or Figure 78 in the appendix). Figure 42 shows the result of the discussion in STWT-4. The driver has to do both: he has to obtain a signature from the customer (*Delivery note as paper form*) and – in order to document the delivery in the mobile communication system – check the goods as delivered in the mask *Signature* on the *Pocket PC*.

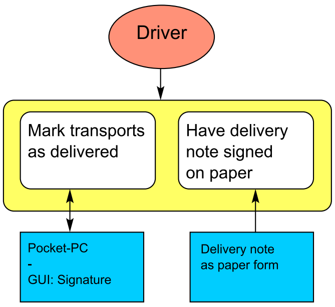

Driver
Mark transports as delivered
Have delivery note signed on paper
Pocket-PC - GUI: Signature
Delivery note as paper form

Figure 42: Usage of Paper Forms

Replacement of other Technical Devices needs to be agreed

A design process concentrates naturally on the artifact it is going to produce. Often however a CSCW-system is deployed to replace some other technical device or mode of cooperation. Example 14 illustrates that such a replacement does not happen automatically. One of the management's goals for the deployment of the mobile communication system was to reduce the dispatchers' workload by reducing the number of conversations by phone with the drivers. If the drivers simply continued to use their mobile phones as before and used SpiW-Com as an additional device for documentation only, this goal could not be met. On the other hand it was obvious that phone calls would still be necessary in addition to SpiW-Com; one could not simply say that mobile phones should not be used anymore.

Figure 40, Figure 41 and Figure 42 give examples of different levels of agreements on the usage of alternative technical devices. While Figure 40 explicitly states under which condition the mobile phone should be used, the examples in Figure 41 leave it to the driver's discretion whether he needs personal contact with the dispatcher or not. The modifier *as little as possible* documents the guiding principle under which the decision is made. In Figure 42 the driver does not have any choice – the customer must sign the *Delivery note as paper form*.

Success and Some Limitations of Discussing Conventions prior to Use

It has been said before (cf. section 3.2.2, p. 86) that planning computer supported work procedures by creating socio-technical self-descriptions neither intends to nor is capable of replacing the processes of evolving use that take place once the CSCW-system has been put to use.

The topic of reducing the usage of mobile phones in case study 1 provides empirical evidence regarding the relationship between "evolving use" and "planning ahead". While it was possible to reach some agreements concerning the usage of mobile phones, we also reached the limits of discussing conventions prior to use. First, consider what has been achieved: After STWT-4 (the qualification workshop) all participants were aware of the fact that the usage of mobile phones is an issue; in fact they shared the understanding (or technological frame in the sense of Orlikowski, cf. section 3.1.2, p. 69) that the CSCW-system SpiW-Com should partially replace the usage of mobile phones. The diagrams created during the workshops include the general intention to reduce the number of phone calls (cp. Example 10, Figure 36); they also document the drivers' choice of selecting the appropriate medium (cp. Figure 41 in Example 14); and there are diagrams that include more specific conditions under which the mobile phone should or should not be used (cp. Figure 40 in Example 14).

Second, consider what was not possible to achieve: it was not possible to define a set of conventions that clearly answered the question whether or not to use the mobile phone for a specific situation.

The limit of planning ahead was reached at this point. The decision whether or not a phone call is more adequate than an entry into the mobile device depends on so many aspects of the specific situation as well as the characteristics of the individual driver that it seems to be impossible and not desirable to prescribe this decision. At this point Suchman's concept of "situated action" becomes effective. The drivers will come to a decision using their own evaluation of the situation; they would not refer to a documentation telling them whether or not calling the dispatcher was appropriate.

Does that mean that the agreements that were reached concerning the usage of mobile phones were useless? No, I argue that much has been achieved with the way the mobile phones are included into the diagrams. In the field study (cf. section 3.1.2, p. 69) Orlikowski and Gash describe how the fact that users did not share basic ideas about Lotus Notes™ lead to the situation that the system was not used for cooperation [Orlikowski, Gash (1994)]. After the STWT-workshops the users of SpiW-Com shared the idea that one purpose of the system is to reduce the number of phone calls and they were all committed to this goal. In section 3.1.2 (p. 69) the development of congruent technological frames was named as one goal for socio-technical self-descriptions in CSCW-projects. The example of mobile phones in case study 1 illustrates that this was indeed possible. This is an important gain for CSCW-projects, in the light of field studies such as [Orlikowski, Gash (1994)] that describe how seemingly obvious concepts were not shared among users, and how this fact eventually results in a reduced usage of the CSCW-system.

7.1.1.6 Meta-level Comments concerning the Project

Example 16: Comments as one way of documenting a Meeting

In STWT-3 of case study „Mobile Communication System" screenshots of the GUI were discussed in context with the diagrams (Figure 74 to Figure 80 in the appendix). The results of the discussion might influence the diagrams as well as in the GUI-prototypes and comments were added at the appropriate places. Figure 43 is a detail of Figure 79 (in the appendix) which illustrates the usage of comments.

Used in this way the comments are places for documenting organizational issues that still need to be decided. The following comment in Figure 43 is a good example: *"In Westkreis, the dispatcher could transfer the tour to the driver when the driver receives the delivery notes; because then one knows which orders he has loaded onto the truck."* This comment resulted from a discussion during STWT-3; a dispatcher and the project-leader discussed what would be the best time for the dispatcher to transfer a set of new tours to the driver. The issue was not settled during the workshop, but the comment documented that the issue has been discussed and what the current ideas for solving it are.

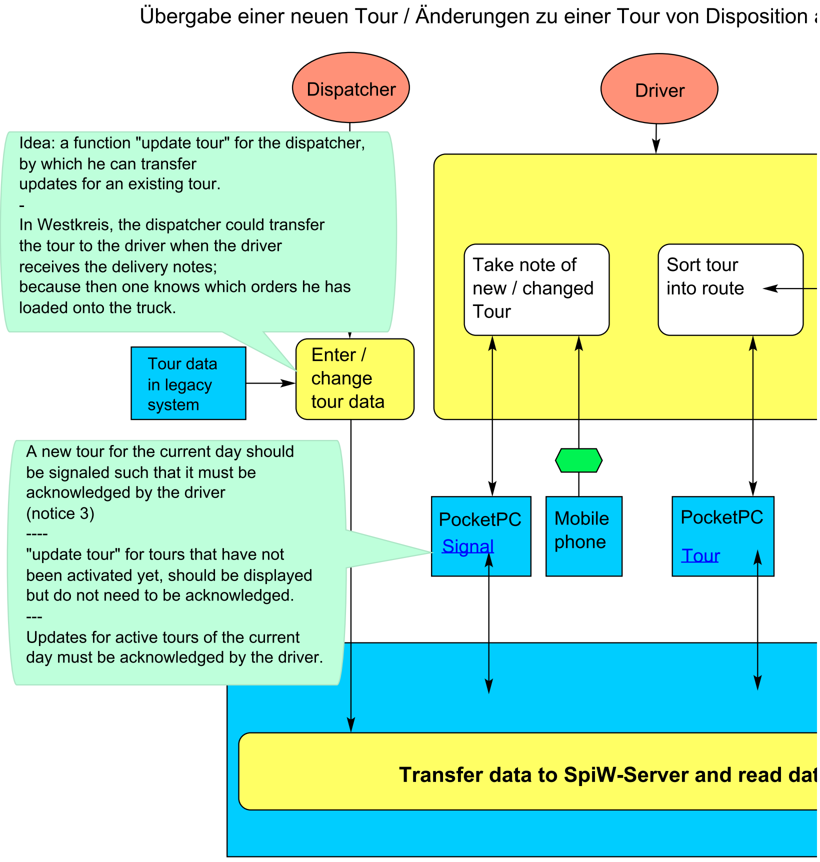

Übergabe einer neuen Tour / Änderungen zu einer Tour von Disposition a
Dispatcher
Driver
Idea: a function "update tour" for the dispatcher, by which he can transfer updates for an existing tour. - In Westkreis, the dispatcher could transfer the tour to the driver when the driver receives the delivery notes; because then one knows which orders he has loaded onto the truck.
Take note of new / changed Tour
Sort tour into route
Tour data in legacy system
Enter / change tour data
A new tour for the current day should be signaled such that it must be acknowledged by the driver (notice 3) ---- "update tour" for tours that have not been activated yet, should be displayed but do not need to be acknowledged. --- Updates for active tours of the current day must be acknowledged by the driver.
PocketPC Signal
Mobile phone
PocketPC Tour
Transfer data to SpiW-Server and read dat

Figure 43: Additional Information Attached as Comments

A Place for documenting Open Organizational Issues

The diagrams are not only used to document decisions reached by the participants, but also issues that remain open. Example 16 as well as Example 8 provides empirical evidence. For a CSCW-project this is a very important aspect, because there is the danger that a topic is considered as settled once all issues relating to the technical system are closed. But often it is much easier to make a technical feature flexible e.g. by allowing different configurations, than waiting for the end of the organizational debate. However these unsolved organizational issues would hinder a smooth deployment of the CSCW-system.

Example 8 describes such an imbalance between the effort necessary to implement a certain software-function and the effort necessary to plan the corresponding organizational procedures. Whether the GUI for the Login contains fields for the driver, the truck or both makes no big difference in the programming and it does not affect the architecture of the system. With regard to the organization however it does make a big difference whether the Pocket PCs are associated with trucks or drivers. This decision influences the number of devices needed, the persons responsible for them, the place they are stored, the place where the batteries are recharged etc. When it turned out during the project that the organization would need more time and discussion on this topic, the software engineers realized GUI-screenshots for the alternatives login by truck or by driver. For the participating software engineer the problem was now off his desk until further notice from the organization: he had delivered both possible options, now it was up to the managers and users to decide what they needed.

In such a situation it is important to keep a reminder that an organizational issue is still unsolved. During the two case studies, the comments within the semi-structured diagrams were used for this purpose: metal-level comments are added to indicate the need for further discussion and decision. The diagram in Figure 34 indicates that a decision needs to be taken by the organization. During STWT-3 this question was discussed and decided, Figure 74 in the appendix resulted from this workshop. The decisions are written down as comments. In case study „Electronic ToC Service" Figure 105 (in the appendix) illustrates a similar example.

7.1.2 C2: Are StSd Distinct from Other Documents?

The second research question concerning the contents of StSd was, whether they really document issues that are new with regard to the documentation commonly used in SWE-projects. Now that the contents of StSd have been systematically described in the previous section, this section can discuss how they differ from other documents. Chapter 4 identified three types of documents that come closest to the initial ideas of socio-technical self-descriptions: use cases in UML (cf. section 4.1.5), scenarios in Carroll' SBD (cf. section 4.3.3, p. 120) and business process descriptions (cf. sections 4.1.6 and 4.3.4). Now that the contents of socio-technical self-descriptions are substantiated by empirical experience, they shall again be compared with those three types of documentation.

Comparing the Contents of StSd and Use Cases

The task *Put truck into roadworthy state* in Figure 28 in Example 2 shall be taken as a case to argue how the content of diagrams for socio-technical self-descriptions differs from that of use cases as defined in the UML (cf. section 4.1.5). The claim is that use cases are likely to either exclude this task from their documentation of the driver's work-procedure. Use Cases are meant to describe interactions with a software system and *Put truck into roadworthy state* contains no interaction with SpiW-Com. The argument cannot simply be that there is no use case description in the world that would include the task *Put truck into roadworthy state*. Of course the information that there are more tasks between *Mark transport(s) as delivered* and *Enter departure from recipient into system* can be included in a textual description by a short remark such as „Enter time of departure when truck is roadworthy again". To elaborate the difference in contents between use case descriptions and socio-technical self-descriptions, the following argumentation is divided into to strands: intention and possibility.

First, the intention of use cases is to completely describe the set of interactions between users and a software application and thereby define the functionality of the software to be designed (cf. section 4.1.5). The intention of socio-technical self-descriptions is to describe the context of use for a CSCW-application such that it can serve as guidance for the organization's members in performing their cooperative work (cf. Term 21). For the implementation of the software it would make no difference whether the step *Put truck into roadworthy state* is included in the documentation or not. The software engineers need to know that there are two functions to be implemented: one to mark a job as completed and a separate one to announce the departure from a customer's site. Within the diagrams and documents of UML it is left to chance how much of the adjoining tasks of the driver are actually documented or not. StSd on the other hand explicitly request a comprehensive description of the work-procedures including the topics summarized in Table 18; and these lie clearly outside the intended scope of use case descriptions.

Second, the possibility of including additional information about cooperative work procedures is given for use cases. Since they are written in plain text, often even without templates, each project can decide what to include into use case descriptions and what not. In [Cockburn (2001)] introductory examples of use cases are given; and for one of them, the author explains [ibid, p. 6]:

> *„The writer added additional business context to the story, illustrating how the computer application operates in the course of a working day. This was practical, as it saved having to write a separate document describing the business process. It confused no one and was informative to the people involved."*

The idea of integrating technical and organizational issues as it is done in this example is in line with the concept of using StSd; and the example can be taken as yet another indicator that there is the need for such a documentation within software projects. What makes the concept of socio-technical self-descriptions unique is that the integrated documentation of organizational and technical issues is systematically foreseen and supported. For use cases this integration lies on the fringes of their purpose, it is possible but not necessary; and here lies the important difference to socio-technical self-descriptions.

A final remark with respect to the UML shall be added: A sequence diagram in UML, which is derived from use cases, could contain an action „repack truck" on the driver's lifeline. But sequence diagrams in UML describe the sequence of interactions between two ore more components (possibly human) on a very detailed level including every message that is passed between them. This misses the purpose of Figure 28 which was to describe work-procedures not to detail interaction sequences.

Comparing the Contents of StSd and Scenarios

Scenarios, especially those used by Carroll (cp. section 4.3.3, p. 120), could in principle include all types of topics listed in Table 18. The two categories of content „work procedures" (cf. section 7.1.1.1) and „relation to software-functions" (cf. section 7.1.1.2) are closest to Carroll's descriptions of scenarios. The two categories containing „agreements" (cf. sections 7.1.1.3 and 7.1.1.4) illustrate how the contents of socio-technical self-descriptions exceed those of scenarios. The difference here is the binding character of agreements documented in StSd. Scenarios are meant as rich examples that trigger the imagination of the participants, they are not meant to be of general applicability. StSd on the other hand are meant to serve as guidance for the organization's members (cf. Term 21) in a number of similar situations. Therefore they include rules to which the participants commit themselves and that are abstract enough to cover a set of situations occurring during the users' work-procedures. The discussion in section 7.1.1 provides Example 14 which includes rules about the usage of mobile phones; or Example 13 which includes the commitment of the library team to trigger the import of ToC each Monday.

This difference in purpose results in very practical differences in the documents: Scenarios are concrete concerning persons and circumstances for helping users and designers to think of the same situation (cf. [Rosson, Carroll (2002), p. 1034] for illustrative examples); rather than using abstract roles such as "driver" and "dispatcher", scenarios would refer to persons by names such as "Hans" and "Karl". StSd on the other hand make use of roles for providing a more general description.

STWT workshops will include the description and discussion of scenarios for invoking the participants' fantasy but then it is always the goal to reach a higher level of abstraction for documentation (cp. Example 37).

Comparing the Contents of StSd and Business Processes in Case Study 1

Similar as for use cases it is impossible to argue that business-process descriptions will never contain the information contained in StSd. In sections 4.1.6 and 4.3.4 business process descriptions and socio-technical self-descriptions are compared based on their conceptualization in literature. In case study 1 the logisticians created business processes in order to derive generalized requirements for supporting IT-infrastructure [Erkens, Kopfer, Siek (2005)]. These provide empirical evidence for discussing the difference between socio-technical self-descriptions and business processes as they were used in case study 1.

Again similar to SWE-documentation, the focus of the two process descriptions is a different one. The inclusion of topics that are essential for StSd is – if at all - of marginal rather than central interest in business process models. Two aspects indicating the differences to StSd shall be shortly discussed. First, the way it is decided whether or not to include a certain activity into the process model. Tasks were included into the business process models if they had a potential impact on the further direction the process would take. They were aggregated to a level where a task either produced a graspable output or where the process forked according to the outcome of the task. Example 2 above is one among many describing that the selection of relevant tasks is done differently for socio-technical diagrams. Here it is important to come to a description suitable for providing guidance for the participants; this may but need not coincide with business process descriptions generated by the rule given above. The second aspect concerns the level of abstraction to which the diagrams were brought. The logisticians' business process models needed to be so abstract that they would support the design of software that would suit the needs of most German logistic companies. The StSd on the other hand were created to support the structural coupling of one organization to one mobile communication system; therefore the information contained in StSd is much more concrete with respect to one organization. Summarizing one can say that the different focus that StSd have, leads to a less abstract description of the work-procedures.

This leads to another observation made while analyzing the diagrams of the case studies. The socio-technical self-descriptions were a good place to document agreements that are concerned with details of the work procedures. The larger picture which is typically represented by business process descriptions is inscribed in the CSCW-system itself. Socio-technical self-descriptions add more detail in order to support the evolvement of viable work procedures. The agreements concerning the usage of mobile phones in case study 1 (cp. Example 14) or the reminders to be sent by the library staff (cp. Example 11) are examples.

Some Thoughts on the Completeness of StSd

How does one know that a socio-technical self-description is complete? The examples in section 7.1.1 show quite different types of actions that were included in the work-procedures although at first sight they did not seem to have a direct link to the software-system (cp. Example 2). Some of them were kept until the end of the project because they served a purpose in the diagrams and the associated communication processes; others disappeared as the project proceeded because they were not needed anymore. The question of granularity arises with respect to modeling interactions with the CSCW-system (cp. Example 4 and Example 5).

Based on the experience from the case studies, the list of topics summarized in Table 18 should always be checked for relevance in any CSCW-project. But then, the completeness of StSd cannot be specified using formal methods. The coordinated use of a CSCW-system within an organization could be an indicator for a useful StSd in that organization. If the CSCW-system is not used or if its usage causes decreases the overall efficiency, information might be missing in the StSd. Thus the smooth functioning of the socio-technical system rather than formal categories helps assessing the completeness and correctness of a StSd.

7.1.3 C3: How did the Topics Emerge?

The third question related to the contents of StSd is how the topics eventually documented, emerged during the project. The relevance of this question is caused by the nature of action research projects. I was actively involved in the projects and if the topics identified in section 7.1.1 were only induced by me, one could argue, that they did not truly connect to the needs of the organization. However the examples given in this section will demonstrate how the topics were derived from an interaction between us and participants.

Sources for Content

The analysis of case study 1 provides the best empirical evidence for this question, because the participants of the workshops were not related to my research group at all (cf. chapter 6 for a discussion of the characteristics of the two case studies). Analyzing the workshops and documents of case study „mobile communication" one can distinguish three ways in which the participants contributed to the contents of the StSd.

1. In the early phases of case study „mobile communication" the contents in the diagrams were determined by the interplay between diagrams prepared by us and add-ons provided by the participants of the workshops. In these phases the participants' contributions remained close to the content already contained in the diagrams. Example 1 (p. 171) illustrates this process.
2. As the workshops continued, new topics were included into the diagrams that spontaneously came up during the discussion. Example 17 illustrates this process for the topic of mobile phones.
3. Finally content was collected by the participants in response to open questions by the facilitator; Example 18 and Example 19 are taken from the qualification workshops and illustrate this way of gathering contents for StSd.

The rest of this section is dedicated to three examples describing how issues that were documented in the StSd were found. Taken together, all examples illustrate that the topics in the diagrams were by no means dictated by us (the involved researchers). They evolved in both projects as the CSCW-projects advanced

Example 17: Who induced the Topic of Mobile Phones?

As one example how topics first mentioned by the participants were included into diagrams, the topic of mobile phones as an alternative to the mobile communication system has been traced to its origins: During STWT-2 one of the dispatchers started the first discussion on mobiles vs. SpiW-Com by stating that he would continue to use the phone in urgent cases (cf. transcript in 11.4.3). A short ironic exchange that the mobile phones would be collected once SpiW-Com was operational followed, but then project leader and software engineer reassured him that SpiW-Com was only a „supporting system" that may be used to reduce the number of phone calls.

From then on the topic of mobile phones was discussed regularly and also included in the diagrams. During STWT-4 the participants brought up the topic again because they felt the need for further arrangements (cp. Figure 44).

Example 18: Need for Arrangements concerning the Use of SpiW-Com

Figure 44 shows a board that was created during the qualification workshop (STWT-4) in case study 'Mobile Communication System'. After the participants had received training in the handling of the mobile devices, the facilitator asked: „Where do you expect further need for organizational measures?" The answers given by the participants were written onto moderation-cards. Table 19 provides a translation with additional explanation

Column	Topic (as written on card)	Explanation
1	What is „rusty"?	The GUI provides standard fields for common problems. One of these fields that can be checked is „material is rusty". The drivers wanted to discuss whether they should check this field only when the material is so rusty that the customer does not accept the goods; or also when there is only little rust that gave rise to some complaints.

Column	Topic (as written on card)	Explanation
	When is the point for the driver to inform the dispatcher?	With the mobile communication system the driver can inform the dispatcher about problems for which he would not phone him but that may be worth noting. The drivers saw a need to discuss the expectations of the management and the dispatchers, what kind of problems should be noted in the system.
	System not as an end in itself	The drivers wanted to make sure that technical features will not be used „only because they are there”; but that the efficient support of their work is in the focus of discussion.
	Meaningful information	Especially when using fields where plain text can be entered, the participants admonished each other to enter only meaningful information
2	Consistent terms	The participants found it necessary to agree to a set of standard terms for certain situations.
	What is the meaning of predefined messages?	SpiW-Com provides predefined messages that can be selected rather than entering free text. The participants wanted to discuss the meaning of those predefined messages.
3	How is information processed?	The drivers wanted to know how the information they enter into the mobile client is further processed.
4	Reliable reaction to information	The drivers wanted to be assured that the dispatchers will reliably react to the information entered into the mobile client.
	Fast reaction by dispatchers	The drivers wanted to make sure that the dispatchers would react fast to their requests entered into the mobile client.
5	When is the time to use the mobile phone?	The participants considered it necessary to agree on situations in which the driver could righteously use the mobile phone.

Table 19: List of open Issues in „Mobile Communication System”

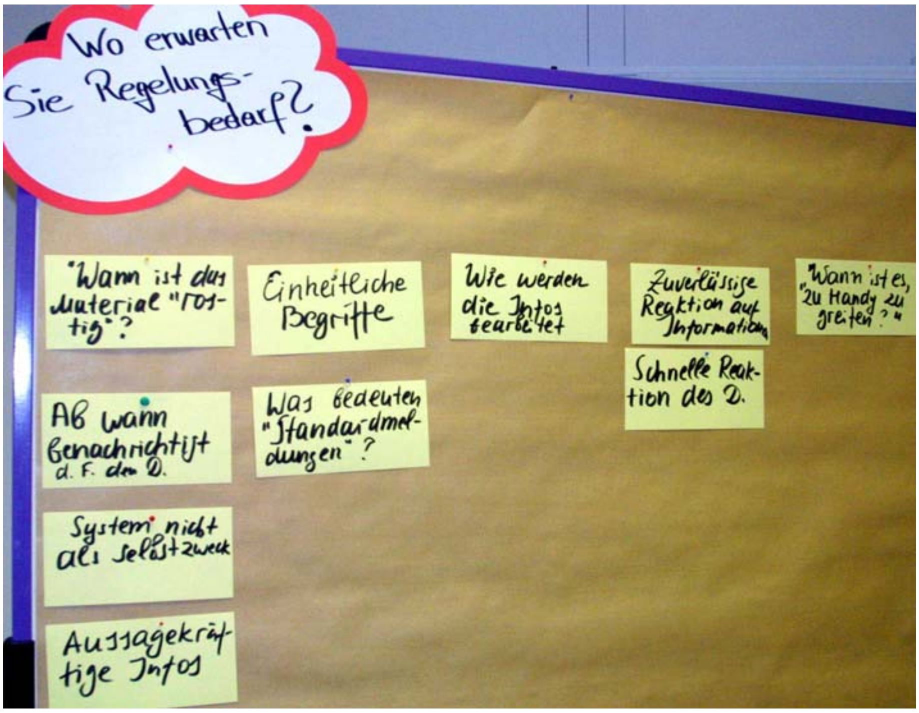

Wo erwarten Sie Regelungsbedarf?
"Wann ist das Material "rostig"?
Einheitliche Begriffe
Wie werden die Infos bearbeitet
Zuverlässige Reaktion auf Information
"Wann ist es „zu Handy zu greifen"?"
Schnelle Reaktion des D.
Ab wann Benachrichtigt d. F. den D.
Was bedeuten "Standardmeldungen"?
System nicht als Selbstzweck
Aussagekräftige Infos

Figure 44: Open Issues in 'Mobile Communication System'

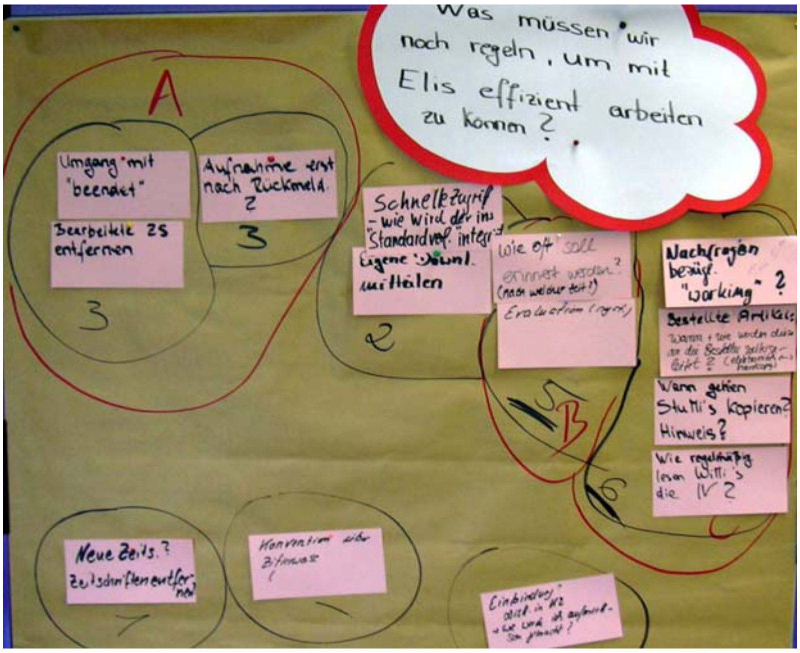

Was müssen wir noch regeln, um mit Elis effizient arbeiten zu können?
A
Umgang mit "beendet"
Aufnahme erst nach Rückmeld.
2
Bearbeitete ZS entfernen
3
3
Schnellzugriff – wie wird der ins "Standardverf." integr.
Eigene Journ. mitteilen
2
Wie oft soll erinnert werden? (nach welcher Zeit?)
Evaluation
Nachfragen bezügl. "working"?
Bestellte Artikel
Wann gehen Stutti's kopieren? Hinweis?
Wie regelmäßig lesen Willi's die IV?
5
B
6
C
Neue Zeits.? Zeitschriften entfernen
Konvention über Bilanz
Einbindung
1
1

Figure 45: Open Issues in „Electronic ToC-Service"

Example 19: Need for Arrangements concerning the Electronic ToC-Service

A similar question was posed in the qualification-workshop in case study „Electronic ToC-Service": „What do we need to organize in order to use Elis efficiently?" The clustered answers are shown in Figure 45. Table 20 contains the translations and further explanations where necessary.

Cluster	**Topic** **(as written on the cards)**	**Explanation**
1 (C)	Requests concerning „working”	The electronic system „Elis” foresees the state „working” when the student workers have started processing a ToC; i.e. they have started to retrieve the articles ordered by the scientists. This cards formulates the scientists’ need to gain intermediate information about the process of acquisition.
	Ordered articles: when and how will these be forwarded to the orderer? (electronic form vs. hardcopy)	
	When do the student workers go and Xerox the articles? Information?	
	How regularly will the scientists read the ToCs?	
2 (B)	How often should reminders be sent? (After how many days)?	
	Evaluation (system)	The idea behind this card is that the software could generate information about the usage of the system. This information could then be used to generate automated reminders or to indicate the need for further organizational measures.
3	Quick access – how will that be included into the standard procedure?	There is always the need for immediately retrieving certain articles because they are important for a research project, a lecture or a publication. This card brings up the question how such a „quick access” can be synchronized with the normal procedures.
	Inform about own downloads	This card refers to the fact that the scientist will sometimes download articles that are important for them on their own. The question is how this information is passed into the regular procedures.
4 (A)	Handling of „finished”	This card brings up the question of what should happen to ToC that have been processed.
	Removal of processed journals	
	Automatic inclusion of articles?	Articles that are used by the scientists in their research or teaching are stored in a literature-database. The question is whether articles selected in a ToC should automatically be transferred into this database or whether the explicit order by a scientist is necessary for this.
	New journals? Removal of old journals?	This card brings up the question of who is responsible for updating the journal database within the system; and what rules should be applied to this work
	Conventions concerning citations?	This card brings up the question whether description of articles in the database should follow certain conventions. The idea is to easily use them for citations in publications.
	Inclusion of discussions in K2 – how am I alerted?	Discussions about articles should take place in the knowledge-management-system K2. The question

Cluster	Topic (as written on the cards)	Explanation
		is how the researchers are informed about the fact that a discussion is going on concerning a certain article.

Table 20: List of Open Issues in „Electronic ToC-Service“

7.1.4 C4: Importance of StSd?

Now that it has been illustrated how the topics contained in the socio-technical self-descriptions were found during the project, the final question related to the contents of StSd is whether the participants attached any importance to the fact that the topics were discussed and documented.

To this end questions regarding the socio-technical diagrams were included in interviews conducted in the scope of the case studies.

Example 20: Drivers' Feedback concerning StSd during Training

The qualification workshop in case study 'Mobile Communication System' consisted of two parts. First, the drivers learnt how to handle the mobile client. For this purpose, the socio-technical diagrams prepared during the course of the project were displayed to explain the context of use for the software (e.g. Figure 30 or Figure 31). They provided a background for the detailed explanation of each function of the software. In an interview a few days after the workshop, the two drivers were asked: „How helpful were the diagrams during the training session for the drivers?" (cf. section 11.5.1.3 for a complete transcript of the question and answers). Both drivers clearly stated that the socio-technical diagrams were an important part of the training.

Driver 1 (HB) first answered „Well yes, that is necessary …". The interviewer then rephrased the question: „Well, we debated ourselves, do we have to do it this way? Maybe the user-interface is sufficient … we were uncertain." Again the driver answered „Those models, they were better. I think it should be done the same way for the next training for the drivers."

Driver 2 (HK) answered „Yes, of course, one has to know, what one has to have in mind, …, when one takes the next step in the software-program."

Example 21: Participants' Feedback after Qualification Workshop

In the interview following the qualification-workshop in case study „Mobile Communication System", the participants were asked how important it was to discuss further rules for the software-system. All participants (drivers, dispatcher, manager, and software engineer) stated that it was very important to do so because otherwise the system could not be used effectively. A few excerpts of the answers are summarized in the following table (cf. section 11.5.1.1 for a transcript of the question and the answers).

Question: „Now, after the workshop, do you consider it necessary and sensible that dispatchers and drivers put up rules for their cooperation with the new software?”	
Project-leader (also taking the role of a dispatcher)	„I regard it as very necessary, because otherwise everybody would interpret and use the system differently.”
Manager from the central office	„I find it quite sensible, because by doing it we have cleared controversial issues in advance.”
Software engineer	„Definitely necessary … the workshop showed that the system does not implicitly cover everything, but that explicit agreements are required.”
Driver 1	„Yes, I would think it is important.”
Driver 2	„That is very important, because otherwise the whole system would not work.”

Table 21: Feedback to Models in Qualification Workshop

Example 22: Participants' Feedback to Content of StSd in Case Study 2

After four workshops in April 2005 the participants were asked questions about the usefulness of the diagrams. All questionnaires contained 1 closed and 3 open questions; the questionnaire for the last workshop included one additional closed question. The questionnaires were sent out by e-mail a few hours after the workshop; the participants answered within a few days by e-mail as well. The number of participants was too small for a statistical analysis; therefore the presentation of the results remains descriptive. The complete set of answers is included in the appendix in section 11.5.2; at this point selected results are presented.

After all of the four workshops the participants had to mark usefulness of the diagrams' contents on a five-item scale ranging from "they helped much" to "they greatly disturbed". All answers were located in the top two items "they helped much" and "they helped". This positive reaction is substantiated by the answers the participants gave to the open question regarding the usefulness of the diagrams' contents. Table 22 includes translations of prominent examples; the source for each quotation is given by referring to the table in the appendix which contains the complete set of answers.

Which information in the diagrams was especially useful / important for you?	
1	"The process related information, which is not described by the 'pure' technical system ELISE and supplied important background information, was especially useful. … what happens when I need literature, for which the library has already set the status to 'working'" (Table 36, 4-April, row 1)
2	"The diagrams were especially useful for seeing the process of usage in advance. It was clear how to use ELISE." (Table 36, 4-April, row 2)
3	"It is good to see the whole process, including the library's work procedures. … With respect to the status of a table of contents the diagram for "late ordering"[29] illustrates their meanings." (Table 36, 4-April, row 2)
4	"I found the information explaining the interplay between library's and scientists' activities useful. This way I could show what the library is doing and explain the consequences for the scientists and vice versa." (Table 36, 4-April, row 3)

Table 22: Feedback to Contents of StSd in Case Study 2

[29] She is referring to the diagram in Figure 103

Positive Feedback by the Participants regarding the Importance of StSd

The feedback given by the participants of the case studies and documented in Example 20, Example 21, and Example 22 is consistently positive. So although in both case studies the initiative for documenting socio-technical self-descriptions came from our side (the side of the researchers), the participants not only actively contributed (cf. section 7.1.3), they unanimously valued them.

A few of the participants' statements shall be emphasized, because they relate to theoretical topics discussed before. The project leader in case study 1 said that he regarded the agreement on conventions necessary because "otherwise everybody would interpret and use the system differently". This insight relates to the topic of technological frames that has been discussed before (section 3.1.2, p. 69; section 7.1.1.5, p. 188). Different technological frames would lead to different interpretations of the system; with his statement the project leader confirms my interpretation in section 7.1.1.5 (p. 188) that it was indeed possible to support the evolvement of congruent technological frames.

Two of the selected statements refer positively to the fact that the conventions documented in the StSd were agreed to prior to use (Manager in Table 21; participant 2 in Table 22). These statements support my interpretations in sections 7.1.1.4 (p. 185) and 7.1.1.5 (p. 188) that the agreement on conventions prior to the usage of the CSCW-system is possible and helpful. Example 47 and the discussions in section 7.4.2 illustrate that conventions agreed to prior to use are indeed kept during use.

Two of the selected statements are an implicit confirmation of the interplay between the three forms of socio-technical self-descriptions illustrated in Figure 16. The software engineer in case study 1 (cf. Table 21) and participant 1 in case study 2 (cf. Table 22) state that additional background information is necessary for embedding a CSCW-system into work procedures.

Integrating heterogeneous Groups of Users

Two of the statements selected from case study 2 refer to another interesting aspect: socio-technical self-descriptions can provide information across distributed work settings. Participants 3 and 4 (cf. Table 22) refer to the fact that they are not familiar with the work procedures in the library and that the discussion of the StSd provided an insight into the way the student workers act in the library. Although it was not mentioned explicitly in the feedback, the fact that drivers and dispatchers could learn about one another's working procedures was important in case study 1 as well. The open issues collected during the qualification workshop (cf. Example 18) show how uncertain the drivers are about the way the dispatchers will use the system. If a CSCW-project includes workshops during which different groups of users can learn about each other's work procedures, this can help increasing the understanding for the overall process supported by the CSCW-system.

These empirical observations are related to the first problem Grudin identifies in his famous paper "Why CSCW Applications fail: Problems in the Design and Evaluation of Organizational Interfaces" [Grudin (1988)]. Grudin analyzed that one problem with CSCW-applications is that often the benefit of their usage is unevenly distributed. Some people will have additional effort in using them while others will benefit in form of reduced effort on their side. The additional effort might be justified by an overall improvement affecting the organization as a whole. But in order for all users to see the overall improvement, they must be able to see the whole picture and understand how the new CSCW-application will be used by different stakeholders. In case study 2, an uneven distribution of effort and benefit cannot be seen. In case study 1 however, there was some discussion whether the mobile communication system will cause additional effort for the drivers, especially as long as the paper work cannot be substituted completely (cf. Example 15). The workshops, especially the qualification workshop where the driver's as well as the dispatcher's interface could be regarded, provided a context for discussing the advantages of SpiW-Com with regard to the organization "Westkreis" as a whole.

Additional Experimental Evidence for the Impact of StSd

In the discussion above the subjective appraisal of the participants lead to the conclusion that the creation and maintenance of StSd are indeed helpful for a CSCW-project. There is additional evidence from a controlled, experimental field study that the participatory creation of socio-technical diagrams positively influences computer supported collaboration using a knowledge management system [Carell et al. (2005)]. In the case study 8 groups of 3 students each had to plan the process of collaboratively writing and reviewing papers for a university course. All groups created a plan within a facilitated workshop; the facilitation of 4 groups was done the conventional way using visualization techniques such as flipcharts and meta-plan boards [Klebert, Schrader, Straub (1987)]; the facilitation of the other 4 groups used semi-structured diagrams and the STWT (cf. section 5.2.3). The study took place in the winter of 2003 / 2004 and it was my role to ensure that the semi-structured diagrams and the STWT were employed in the same way as in the two case studies presented in this thesis. And although the setting of the experimental study was computer supported collaborative learning (CSCL) rather than CSCW, the conditions are so close that it seems legitimate to transfer the results. The students planned their usage of LiveLink™ which is a commercial knowledge management system and three results which do not pertain to learning but to computer supported cooperation shall be discussed at this point. In comparison with the other groups, students who created socio-technical models for their computer supported cooperation

1. Reached more agreements about the usage of LiveLink™ (e.g. agreeing on the usage of a discussion forum);
2. Discussed and documented more socio-technical issues (e.g. appointing a moderator for a discussion forum);
3. Made more intensive use of LiveLink™ when writing and reviewing their papers.

So the positive subjective statements the participants in my case studies made concerning the importance of StSd are backed by the outcome of a controlled, experimental field study. It is also interesting to note that 22 students evaluated the planning of their computer supported cooperation as helpful, which also confirms the feedback collected in my two case studies.

7.2 Form

The initial concept for supporting StSd in CSCW-projects foresees the usage of semi-structured diagrams as a means of documentation (cf. section 5.2.2); the following sections discuss the experience made with semi-structured diagrams in the course of the two case studies.

7.2.1 F1: Are Semi-structured Diagrams a suitable Form of Documentation?

The first form-related research question asks whether semi-structured diagrams are a suitable form of documentation for the topics that came up during the case-studies. The discussion of this question starts with the observation that it was indeed possible to capture the topics that were considered as relevant by the participants.

Topics brought up by the Participants could be documented

All the examples given in the previous sections show, how topics that became relevant in the course of the two case studies were documented in form of semi-structured diagrams. Now one could argue that only those topics were documented that were suitable for the form of semi-structured diagrams; i.e. that the form of documentation selected the contents. To state a case against this assumption Example 18 and Example 19 shall be revisited at this point.

Example 18 describes how the participants of the qualification workshop in case study 1 listed open issues that needed discussion and clarification. Figure 44 and Table 19 provide lists and explanations of these issues. Due to time restriction not all of the issues could be handled within the workshop; the issue of mobile phones was selected to be pursued in the workshop. One result in diagrammatic form is shown in Figure 84 (in the appendix). Example 19 describes the same procedure for case study 2. In this case however the participants worked in small groups on several topics at the same time; the models in Figure 104 – Figure 107 (in the appendix) are the results.

Thus Example 18 and Example 19 can be taken as an indication that semi-structured diagrams are indeed capable of capturing issues that are relevant for StSd.

Example 23: Dispatchers' Work-Procedures

Figure 46 which is a detail of Figure 64 (in the appendix) shows how photos were used to enrich the information of the semi-structured diagrams. The diagram illustrates the dispatchers' work-procedures as they were documented in the analysis phase of the project. It would have been possible to add further details to the entities *Dispatch notes etc. unsorted in printer* and *dispatch notes presorted according to tour day*, but the photos added much more context for the participants who could thereby better relate the symbols of the diagram to their actual work environment.

Interweaving Semi-structured Diagrams with other Materials

Whenever possible, semi-structured diagrams were mixed with other artifacts that were produced during the design process. In Example 23 photos illustrate the work environment. Another example can be seen in Figure 50; here a photo of the daily record was included into the diagram. It would have been quite easy to translate the lines and columns of the table into entities and subentities in the SeeMe diagram, but the photo transports much more information. Furthermore two forms with similar names ("Tagesbericht", "Tagesabrechnung") existed, and they were mixed up by some of the participants. With the photo it was very clear which form was meant.

Besides photos, other documents such as a list of status values that a tour could take or a list with standard texts that should be used were also integrated into the semi-structured diagrams by using links rather than translating them into symbols of the diagram (cp. Figure 73 in the appendix).

Thus using semi-structured diagrams as the main mode of documentation does not mean that every piece of information has to be translated into diagrammatic symbols. Especially information that is generated and discussed by a different group of people and serves as input into the project is a candidate for being referenced rather than modeled. If such information were translated to be part of the diagram, the diagram would have to be changed each time the external information changes; this topic is discussed again in section 7.2.2 for the representation of the technical system.

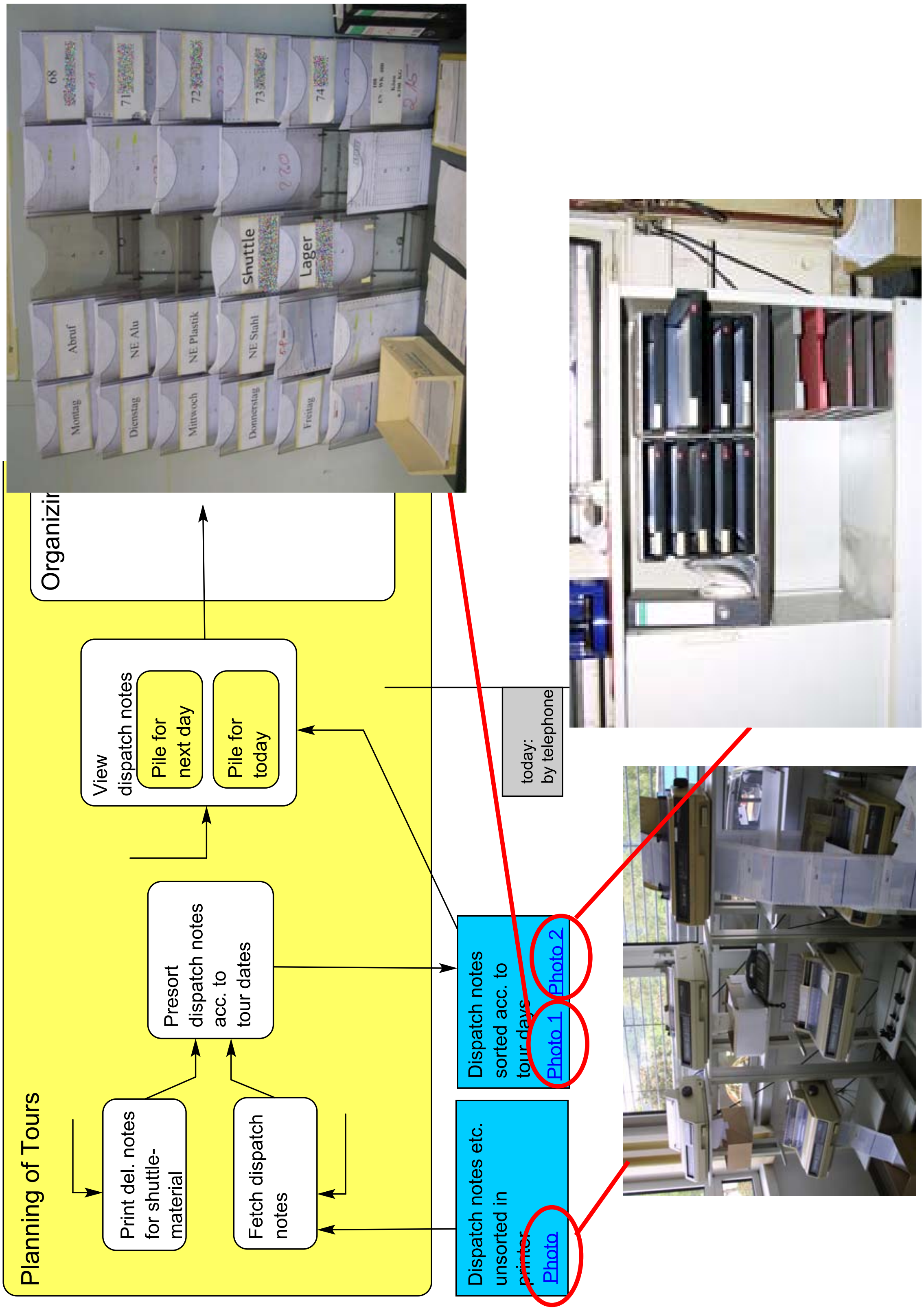


Figure 46: Dispatchers' Work-Procedures illustrated with Photos from Fieldwork

Example 24: Tables were the better Form for Documentation

After the intended work-procedures using the electronic ToC-service had been modeled in case study 2, the task was to select a software system that would support them best. Discussion brought the alternatives down to five rather quickly:

1. BSCW which is a groupware tool developed by Fraunhofer FIT in Germany
2. Filemaker which is a commercial database tool
3. K2 which is a CSCL-environment developed, used and evaluated within the research team
4. LiveLink® which is a commercial knowledge-management tool
5. Newly developed web-based solution

The goal of STWT-4 was to select one (or at most two) of these five. The initial idea was to take the models which document the intended work-procedures and create a new diagram for each technical solution, showing how the procedures would be supported by the software. Figure 96, Figure 97, and Figure 98 (all in the appendix) show the diagrams created for Filemaker, K2 and LiveLink. According to the insights discussed in the previous section, screenshots and prototypic implementations of the different solutions were added to the diagrams. In the workshop they were presented and discussed. However in the course of the discussion it turned out that the differences between the solutions could not really be attached to single steps in the intended work-procedures. The advantages and disadvantages of each software could be documented much easier and clearer in form of keywords on moderation cards. Figure 47 shows the result of the discussion.

Two types of arguments could be distinguished: those that related to steps in the work-procedures and those that were of more general nature.

Advantage / Disadvantage relating to work-procedures	**Advantage / Disadvantage referring to other dimensions**
Support for library tasks	easy to learn
(no) support of acquisition of material	Knowledge for further development available within the group
Function for continuous reminding	interesting for further research
	awareness features

Table 23: Selection of Keywords from Figure 47

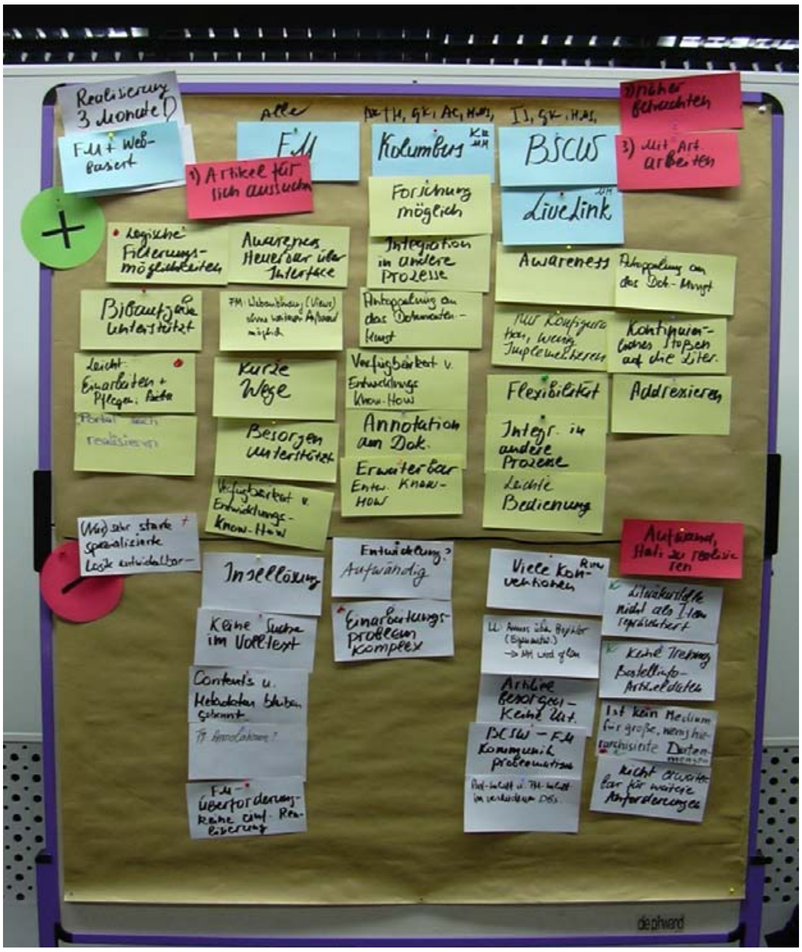

Realisierung 3 Monate!
FM + Web-basiert
Alle
FM
Kolumbus
BSCW
LiveLink
1) Artikel für sich aussuchen
2) näher betrachten
3) Mit Art. arbeiten
+
Logische Filterungsmöglichkeiten
Awareness Hersteller über Interface
Forschung möglich
Integration in andere Prozesse
Awareness
Ankopplung an das Dok.-Mngt
Bildaufgabe unterstützt
Ankopplung an das Dokumenten-Mngt
Kontinuierliches Stopfen auf die Liter.
Leicht: Erarbeiten + Pflegen; Portal nach realisieren
Kurze Wege
Verfügbarkeit v. Entwicklungs Know-How
Flexibilität
Addressieren
Besorgen unterstützt
Annotation am Dok.
Integr. in andere Prozesse
Erweiterbar Entw. Know-How
Leichte Bedienung
Verfügbarkeit v. Entwicklungs-Know-How
–
(Wird) sehr starke spezialisierte Logik entwickelbar –
Inselläsung
Entwicklung? Aufwändig
Viele Konventionen
Aufwand, Hal. zu realisieren
Keine Suche im Volltext
Einarbeitungsproblem komplex
Literaturliste nicht als Item repräsentiert
Keine Trennung Bestellinfo-Artikeldaten
Contents u. Metadaten bleiben getrennt
Artikel besorgen – Keine Unt.
BCSW – FM Kommunik. problematisch
Ist kein Medium für große, wenig hierarchisierte Datenmengen
nicht erweiterbar für weitere Anforderungen
FM – Überforderung – keine einf. Realisierung
dephvard

Figure 47: Pin board with Pro and Con for Technical Solutions in Case Study 2

Some Limitations of semi-structured Diagrams in Workshops

Semi-structured diagrams that are intended to support the structural coupling between an organization and a CSCW-system need to focus on the actions of the users in relation to the CSCW-system. Example 24 illustrates some restrictions that are caused by this focus. The real differences between the technical alternatives in case study 2 could not be described very well by different work-procedures. All of them supported the desired process more or less (otherwise they would not have been among the remaining five options). The majority of keywords listed in Figure 47 and summarized in Table 23 are outside the scope of the StSd as the description of technically supported work-procedures. Although most of the keywords could have been attached to the diagrams (e.g. in form of attributes to entities) this would not have been an effective way to document the arguments.

At the end of the workshop the participants were asked whether the diagrams had helped for the discussion documented by the inboard of Figure 47. Most of the participants agreed that the result could have been reached without the diagrams as well. Some participants however stressed the role of the models as background material for the discussion and for the preparation of the workshop. It is likely that the diagrams did indeed provide a common ground for the discussion. In fact, the same diagrams and their successors were valued highly by the participants in the further course of the project, once the decision for a Filemaker® had been taken and it was necessary to organize cooperative work procedures.

The lesson to be learnt from this example is that when the dimension or level of abstraction of issues does not match the level of activities in work-procedures, an alternative method of documentation should be used.

Another limitation of semi-structured diagrams became obvious during this workshop of case study 2: they are not an appropriate method to describe a large space of options. In this phase of the project it was important to describe the full range of possibilities for each technical option. The diagrams provided for the workshop documented one way in which each technical option could support the intended work-procedure. This was necessary to document the feasibility of the technical options but it was not sufficient to demonstrate their full potentials. One could argue that the diagrams were simply badly designed and that other diagrams could have done the job. However this seems doubtful: the number of options is just too large. It is not simply the question of representing one or two alternative branches of a process. The whole layout of the diagrams would be different for different options; and this makes a comparison difficult.

The conclusion is that for documenting general features leading to a large number of options, semi-structured diagrams are not the best form of documentation. Conventional means of facilitation can be superior. However as a documentation that provides a common ground for the discussion semi-structured diagrams are helpful again; also when it is necessary to check in detail what the influences of a concrete technical option are for the work-procedures.

Example 25: Graphical Details representing important Organizational Issues

There are some examples in the case studies where minimal graphical details represent „big" organizational issues. „Big" can mean that they were intensely discussed or that they imply other organizational or technical changes. Figure 48 (detail from Figure 78 in the appendix) and Figure 49 (detail from Figure 84 in the appendix) illustrate the point.

In Figure 48 (which is a detail from Figure 78 in the appendix) the double ended arrow from *Get information* to *Pocket PC* indicates that the driver not only reads information from the device but can also modify it by entering data. This represents the option that the driver can enter corrections to the recipients' data any time he needs to do so. This option implies that SpiW-Com must provide means to handle the input and it implies that the data entered by the driver must be further processed by the dispatchers.

In Figure 49 (which is a detail from Figure 84 in the appendix) the direct arrow from *Driver* to the subtask *Confer on next steps* implies that the driver is not responsible for initiating the overall conversation by phone. This is the dispatcher's job as indicated by the arrow from *Dispatcher* to the overall process. The diagram is the result from a long and intense debate during the qualification workshop whether or not the driver should call the dispatcher or enter any request into the mobile device.

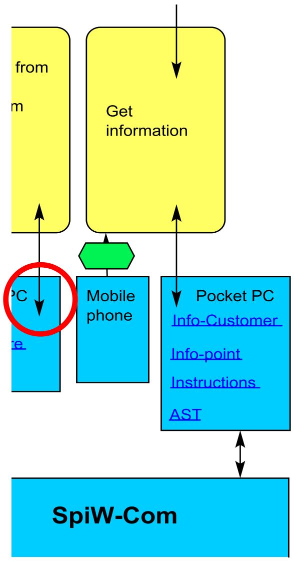


Figure 48: Read only vs. Write access

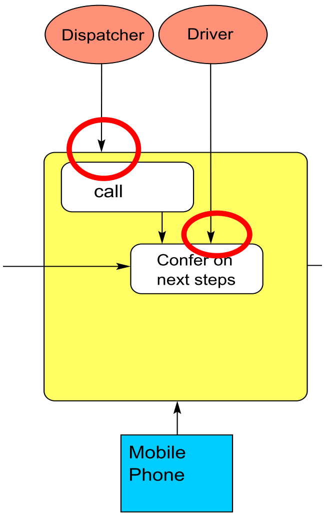


Figure 49: Dispatcher initiates Call

Emphasizing important Issues in Semi-structured Diagrams

The formal aspects of semi-structured diagrams allow the designer to represent aspects of reality in form of brief symbols. This is basis of the capability of graphical representations to provide an overview that can easily be perceived. In [Loser (2005), pp. 143] it is documented that users are able to understand, use and create semi-structured diagrams even with little training.

Example 25 shows the limitations. Figure 48 quasi hides the controversial issue whether or not the drivers are allowed to enter data into the system. Interviews analyzed in [Menold (manuscript in preparation)] showed that the participants of the qualification workshop interpreted the diagram of Figure 49 differently although in general they did understand the diagrams and gave positive feedback to their usage.

The lesson to be learnt from the example is that the non-formal options of semi-structured diagrams should be used to put emphasis on certain aspects. The briefest, most efficient diagram is not the one that necessarily is the best to document a socio-technical self-description. In Figure 48 it would have been quite easy to explicitly add an activity *enter corrections*. Similarly Figure 49 could be redrawn in a way that the activities *call* and *Confer on next steps* are clearly separated with the participating roles directly attached.

This discussion is related to the discussion in section 7.1.2 (p. 194) about the completeness of socio-technical diagrams. Semi-structured diagrams serving as StSd need to adequately reflect the communication processes within the organization; their correctness and completeness is determined by the degree in which they are able to serve this purpose. Although it is syntactically correct and correctly describes the agreement, Figure 49 is not a good example for an StSd because it is not able to adequately document the self-descriptions elaborated during the workshop.

Example 26: Conflict with other Modeling Notations in Case Study 1

There was one organizational problem encountered in the case study „Mobile Communication System" that pertains to the usage of graphical representation of work-procedures. The logistics company to which the project „Westkreis" belongs has been using formal graphical models to represent their workflows and business-processes for two purposes:

a) They were the basis for analyzing and optimizing their business

b) They were the basis of the ISO-Certification of the company's business processes

The problem was that the head of quality assurance who is responsible for the creation and maintenance of these models simply refused to allow another modeling technique to be used in the head office. She feared the two types of models would compete with each other. The solution for the case study „Mobile Communication System" was resorting to the project „Westkreis", and making it clear that the models are used for the purposes of the case study only.

Problem of Competing Modeling Methods within one Organization

In Example 26 two forms of (socio-technical) self-descriptions with overlapping contents became relevant within one project. That this can happen is no surprise, section 3.1.3 discusses the different forms in which (socio-technical) self-descriptions are found in organizations; and section 4.3.4 discusses business process models in particular.

There are two alternatives for handling this problem:

a) The two forms are kept apart because different purposes justify different documents. Links between them are added where appropriate.

b) The two forms are merged because they are similar enough to do so.

In the case study only solution a) was tested, and there are arguments that this is a sensible way to proceed. It is their purpose that makes socio-technical self-descriptions in the context of CSCW-projects special: they are written artifacts that develop from a process of organizational change and that exist to make oral communications sustainable. Models of business processes especially when they are used for receiving ISO-certifications must adhere to certain rules regarding their formalism and content; and this could render them awkward to use as self-descriptions where it is up to the participants to decide what needs to be documented in what level of detail.

Option b) should only be considered if the diagrams can be used freely for the purposes of supporting the structural coupling within the CSCW-project. The arguments given in section 4.1.6 and 4.3.4 related to the theoretical base of creating and using models in organizational change apply here. Section 7.4.1 discusses the potential and limitations for socio-technical self-descriptions to function as boundary objects and thereby be meaningful to more than one group of persons.

StSd in the Light of a skeptical View on Representations of Work

The suggestion of using graphical representations of work-procedures touches a sore spot in the CSCW-community and therefore needs some reflection. Section 4.3.5 contains a summary of the arguments for and against artifacts representing work-procedures. The following section analyses how we tried to exploit their advantages while avoiding their potentially harmful sides at the same time.

One of the claims against process-descriptions is that they reduce the abundance of options in every-day working live to deterministic, linear procedures. When looking at the process diagrams created in the course of the case studies, one has to admit that they simplify the work-procedures. But one also sees that they do not force actions into sequential orders where this was not possible. Figure 46 is a detail from diagram Figure 64 (in the appendix) and illustrates the dispatcher's actions while planning the tours for the next day. Every once in a while he *fetches* new *Dispatch notes* from the printer at the end of the hall and he also prints out delivery notes for the so-called shuttles. There is no schedule to this. The dispatcher alone decides how often and when he gets those papers and includes them in his workflow.

The diagram in Figure 46 illustrates this way of working by the arrows that lead to the tasks *Print dispatch notes for shuttle material* and *Fetch dispatch notes*. They both come from nowhere – they have an open beginning indicating that there is no defined predecessor for this task and that it is up to the person working to decide when to come to these tasks. The same explanation applies to the arrow leading into the task *View dispatch notes* which also has an unspecified beginning.

The diagram in Figure 31: Rough Cluster of Actions from Case Study „Mobile Communication" is another example for avoiding the implication of determinism where there is none. It shows some tasks that are necessary for reaching the right place to unload the truck. But it does not put them into a chronological order and it explicitly states that there might be more tasks to do by the semi-circle at the bottom of the yellow process symbol. Also, the diagram describes the usage of functions of the mobile communication system only vaguely: a list of functions (*AST*, *Instructions*, *Info-Customer*, *Info-Loading-Point*) is given. But the diagram does not imply that there is a necessity for the driver to use any of these functions for a specific task: the arrow connecting the yellow process-symbol with the blue symbol for the *Pocket PC* cuts both edges indicating that all or part of the technical system is used by all or some of the tasks within the process.

By using a semi-structured modeling language the pretense of determinism that does not exist in real life can be avoided.

Another argument that is used by several authors is that the usage of process descriptions becomes dangerous when they are taken away from the context of their origin and when the recipients take them for an objective account of reality. This potential danger cannot be overcome by the properties of the diagrams themselves but rather by methods that make use of the diagrams. The diagrams in both case studies were used to document the organization's socio-technical self-description which was the (intermediate) result of a participatory design process. Their primary purpose is to support the task of structural coupling that the organization has to undertake. For this the diagrams remain within their context of origin.

The semi-structured diagrams should however serve the purpose of boundary objects between different group of users and between the users and the software engineers. But also here the usage of the diagrams is accompanied by facilitated workshops. Thus by embedding the diagrams into a communicative context it is avoided that they are mistaken for an objective account of reality that used in the way construction plans are used for technical artifacts (cp. 7.3 for details).

The usage of the graphical models in both case studies concentrated on the two potentially helpful functions that are given in Table 10: first, graphical representations can support the process of discussing and changing work-procedures; second they can support the coordination that is necessary for cooperative work-procedures.

So far I discussed the adequacy of semi-structured diagrams from my perspective as a researcher. The following two examples summarize the participants' feedback.

Example 27: Feedback to Diagrammatic Form in Case Study 1

The participants of the qualification workshop in case study 1 were asked whether they thought that semi-structured diagrams were a suitable means to document the outcome of the workshop. All of them agreed that this was the case. Some statements from the interviews are summarized in Table 24; the complete transcripts are included in the appendix, section 11.5.1.2. There were some skeptical remarks concerning the usage of the diagrams to aid the facilitation of the workshop; these are presented and discussed in section 7.3, Example 40.

Project-Leader / Dispatcher	„When I look at it after a workshop, the systematic of the models are much easier to understand; the same would be true for an outsider …"
Software engineer	"A picture says more than 1000 words. Therefore the modeling is helpful. … For each method there are advantages and disadvantages. … The advantage [for SeeMe] is that the notation can describe weakly structured work procedures - that is obvious. But that induces the disadvantage: certain analyses that are based on certain formalisms cannot be done."
Driver 1	"I quite liked that [documentation]; it was helpful for me to get a better, even better picture of the system, to become more familiar with it. Because it became more complicated, it got more and more."

Table 24: Feedback concerning Diagrammatic Form in Case Study 1

Example 28: Feedback to Diagrammatic Form in Case Study 2

Feedback in concerning the form of semi-structured diagrams was collected in case study 2 using one closed and one open question. The complete feedback is included in the appendix in section 11.5.2 in Table 37 and Table 39.

The closed question was "How do you judge the usefulness of the form of the diagrams (i.e. SeeMe Diagrams) of today's meeting?". The participants had to rate the usefulness on a five-item scale ranging from "it helped very much" to "it greatly disturbed". All answers lay in the top two categories "it helped very much" and "it helped".

In the open question the participants were asked whether they know other means of documentation that they think would have been better. With the exception of one participant in one workshop (26-April, row 5, in Table 37) the participants unanimously stated that the form of semi-structured diagrams is a good form for documentation. There were some suggestions for improvement which include the additional usage of text and more extensive usage of screenshots. Also worth noting are several remarks concerning the bad quality of the projection and the need to jump between diagrams because they were too large.

Positive Feedback Concerning the Form of Documentation

In sum, the feedback summarized in Example 27 and Example 28 confirms the positive evaluation of semi-structured diagrams as a means to document socio-technical self-descriptions. In case study 2 I deliberately asked the participants whether they had preferred other forms of documentation, because I could expect that they knew about different methods and could weigh the alternatives. The fact that the majority of these participants stated, that they would not prefer other methods, is an additional confirmation. But there are some issues worth discussing.

First, the feedback shows that all participants need a passive knowledge of the modeling language. The participant, who gave the negative feedback in case study 2, and the driver in case study 1, who mentioned that the diagrams added complexity, were the ones with the least knowledge of the modeling notation. Section 7.3.1 includes discussions on the cognitive effort that modeling costs (p. 229) and the requirements regarding the knowledge of the modeling notation (p. 233).

Second, the feedback confirms the idea of enriching the diagrams by interweaving other material such as screenshots. In those cases where this was not done, the participants made corresponding remarks in their feedback (e.g. 4-April in Table 37).

Third, the problems concerning the bad quality of the projections and the problems with the visibility of large diagrams can be solved by providing additional handouts (which we did in STWT-3 for case study 1). But more interesting for further research is the option to solve these problems by improving the technical support for working with diagrammatic models. Section 7.3.1 (pp. 234) includes suggestions for improving the technical support.

7.2.2 F2: Representation of Technical System

So far the emphasis lay on the representation of work-procedures and their relation to the CSCW-system. The question discussed in this section is how the software-system is represented in socio-technical diagrams.

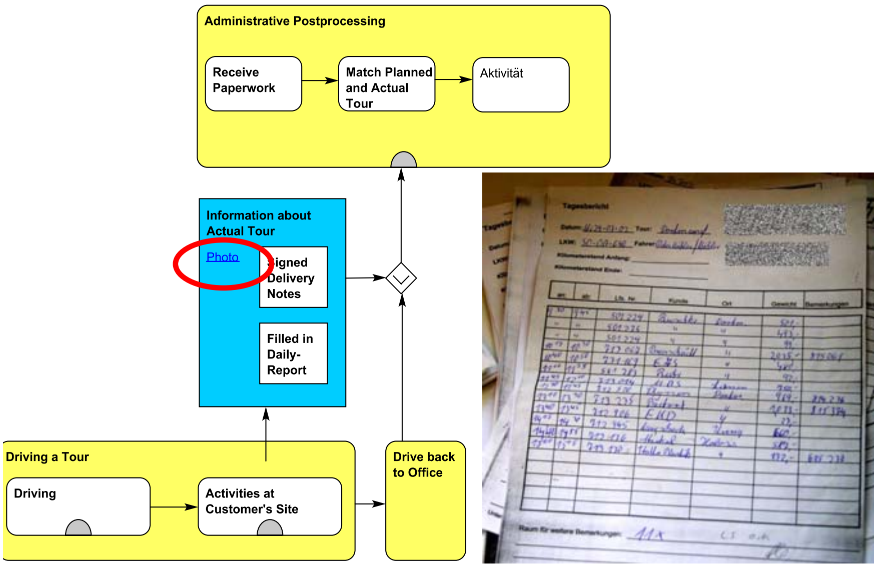


Figure 50: Description of Daily Record in Workshop STWT-1

Example 29: Representation of „Daily Record" throughout the Project

The way in which the CSCW-system is represented in the StSd changed in the course of the projects in both case studies. This example follows one topic through the whole project and illustrates the changes in its representation.

Figure 50 to Figure 53 show how the same topic – the form „daily record" in case study „Mobile Communication System"– is represented in the diagrams in various stages of the project. Figure 50 is a translation of a detail of Figure 62 in the appendix; it shows that the driver has to add data to a form called „daily record" each time he arrives at a customer's site, and that at the end of his tour he takes the form back to the office where it is needed by the administration to bill the tour. Rather than using notational symbols for describing the form „daily record" in detail, a photo of a „daily record" is linked to the diagram, providing the participants with more plentiful information.

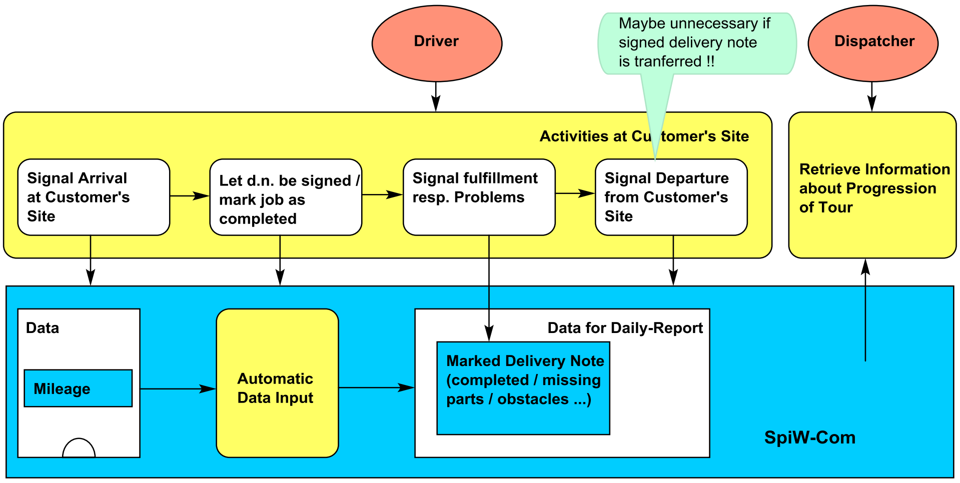
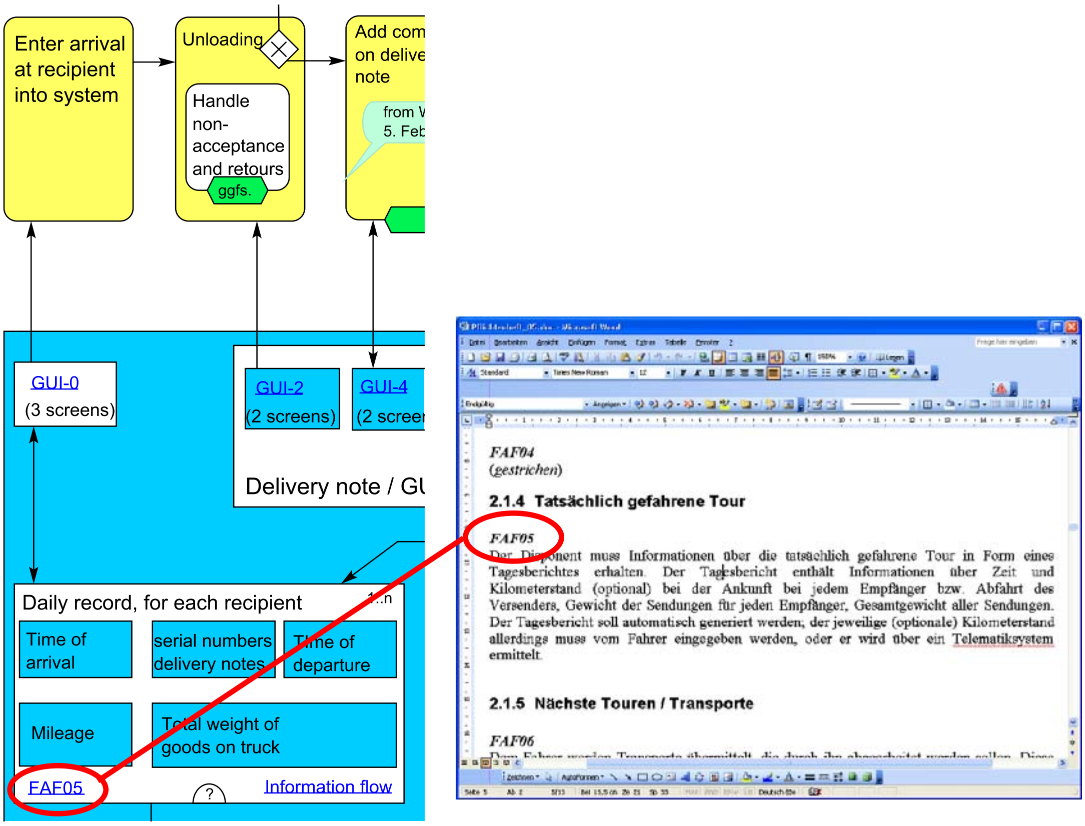

Driver
Maybe unnecessary if signed delivery note is tranferred !!
Dispatcher
Activities at Customer's Site
Signal Arrival at Customer's Site
Let d.n. be signed / mark job as completed
Signal fulfillment resp. Problems
Signal Departure from Customer's Site
Retrieve Information about Progression of Tour
Data
Mileage
Automatic Data Input
Data for Daily-Report
Marked Delivery Note (completed / missing parts / obstacles ...)
SpiW-Com

Figure 51: Description of Daily Record in Workshop STWT-2

In the next phase of the project the participants of workshop STWT-2 were asked to envision how SpiW-Com could support their work-procedures. In this phase no artifacts to link to existed, so everything was modeled using the notational elements of SeeMe. Figure 51 (translation of Figure 67 in the appendix) shows one diagram produced during this workshop: A function *Automatic Data Input* in *SpiW-Com* should automatically collect certain data such as the *mileage*; the *Driver* would add data as he proceeds with his business. Both would lead to *Data for Daily*-Report. The *dispatcher* can *Retrieve Information about Progression of Tour* any time he needs to.

For the software to be developed within the research project SpiW the requirements from 'Westkreis' were only one of several inputs. During the time that followed workshop STWT-2 additional documents were generated and lead to a requirements document. In order to verify their compliance with the ideas for 'Westkreis', the diagrams were extended to include links to these documents. Figure 52 is again concerned with the daily record; it contains a detail of Figure 73 (in the appendix) which was created between STWT-2 and STWT-3. The data to be collected for the daily record is modeled and links to two further documents are included: FAF5 links directly to the paragraph in the requirements document that discusses the daily record; and *Info flows* relates to an additional document describing the information needed mainly from a logistician's point of view



FAF04
(gestrichen)

2.1.4 Tatsächlich gefahrene Tour

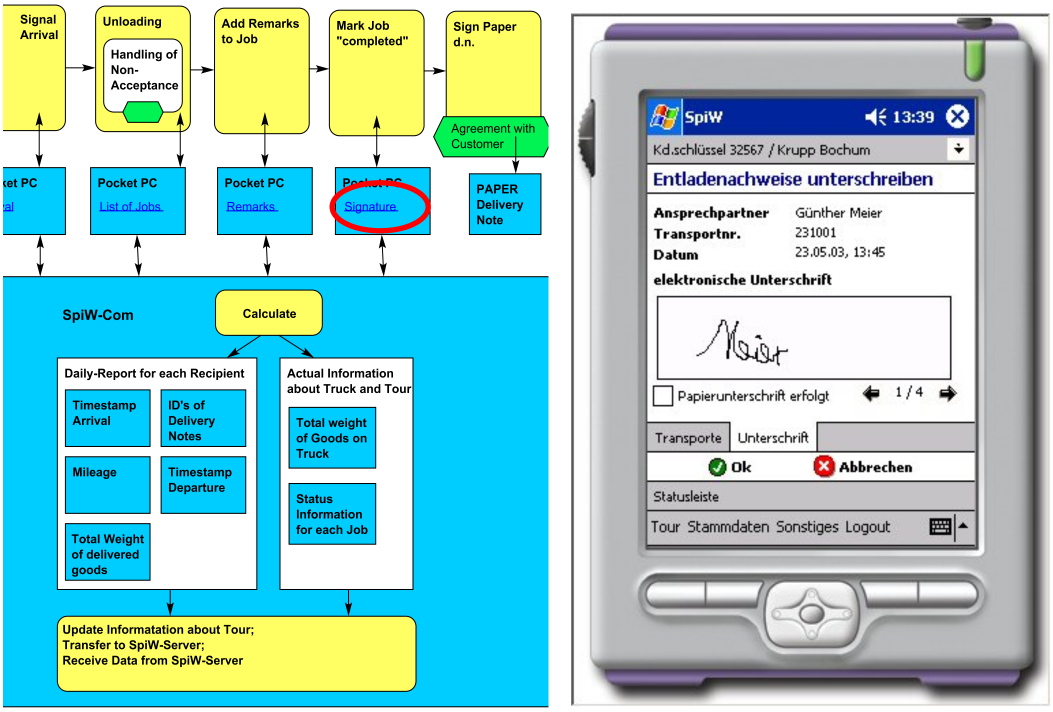
FAF05
Der Disponent muss Informationen über die tatsächlich gefahrene Tour in Form eines Tagesberichtes erhalten. Der Tagesbericht enthält Informationen über Zeit und Kilometerstand (optional) bei der Ankunft bei jedem Empfänger bzw. Abfahrt des Versenders, Gewicht der Sendungen für jeden Empfänger, Gesamtgewicht aller Sendungen. Der Tagesbericht soll automatisch generiert werden; der jeweilige (optionale) Kilometerstand allerdings muss vom Fahrer eingegeben werden, oder er wird über ein Telematiksystem ermittelt.

2.1.5 Nächste Touren / Transporte

FAF06

Figure 52: Description of Daily Record with Reference to Requirements Document

The software-developers provided GUI-prototypes for all workflows that SpiW-Com supports. In order to represent the GUI-prototypes together with additional information about the work-procedures, new entities labeled *Pocket PC* were included into the diagrams; each of them contains a link to a set of related GUI-screenshots. Figure 53 (translated detail from Figure 78 in the appendix) describes the driver's task to fill in the „daily report" by referring to screenshots of the GUI which are to be used for this purpose. Figure 53 shows the mask used fort he *signature*. In addition this Figure 53 provides an example how descriptions for automated functions performed by the software are represented within SeeMe diagrams: SpiW-Com automatically *calculate*s certain figures such as the *total weight of goods on truck* when the driver enters his data about the delivery.



Kd.schlüssel 32567 / Krupp Bochum

Entladenachweise unterschreiben

Ansprechpartner Günther Meier
Transportnr. 231001
Datum 23.05.03, 13:45

elektronische Unterschrift

□ Papierunterschrift erfolgt 1 / 4

Figure 53: Description of Daily Record in STWT-3

Overlap and Need for Adjustments in Representation

Two effects eminent in the previous example shall be discussed in this section: first the overlap between StSd and technical documentation for software-development; and second, the need for StSd to constantly adapt to the status quo of software-development.

One has to keep in mind that socio-technical self-descriptions and technical documentation for software engineering pursue different goals. While the former is an artifact to support the members of an organization in planning their technically supported work-procedures, the latter is a constructional artifact to guide the design and implementation of software. Nevertheless in representing aspects of the software, they overlap. StSd include representations of the software to be able to describe the structural coupling; documents created during the course of software engineering describe the software in order to specify its details for implementation.

StSd should belong to that category of documents that describe the software-system from the user's perspective. In this respect they are comparable to use cases, user requirement specifications, user manuals or acceptance test manuals. The difference is that StSd are not limited to the description of the software-system but include the organizational context from the very beginning.

And this brings up the second issue: while the description of the organizational context keeps its appearance as semi-structured diagrams from the very beginning to the end of the project; the software is first described by graphical or textual means which are then transformed into the program itself. Therefore the links of work-procedures to a CSCW-system within a StSd must be updated in the course of the project.

It is said above that StSd describe the technical system on the same level as user requirements documents do; a valid question is whether StSd could substitute SWE-documents such as a requirements document. In case study „Mobile Communication System" this was not the case; but in case study „Electronic ToC Service", the semi-structured diagrams did indeed serve as requirements documentation.

Example 30: Semi-Structured Diagrams as Requirements Documentation

The diagrams in Figure 92 to Figure 96 in the appendix show how SeeMe Diagrams were used to refine the requirements for the software until they finally describe the contents required in the database of the case study „Electronic ToC-Service". Figure 54 is a detail from Figure 96 (in the appendix) and shows the data fields and automatic functions required from the Filemaker® system. In this case study no further requirements document was written, the SeeMe diagrams were used to elaborate entity-relationship models for the database design.

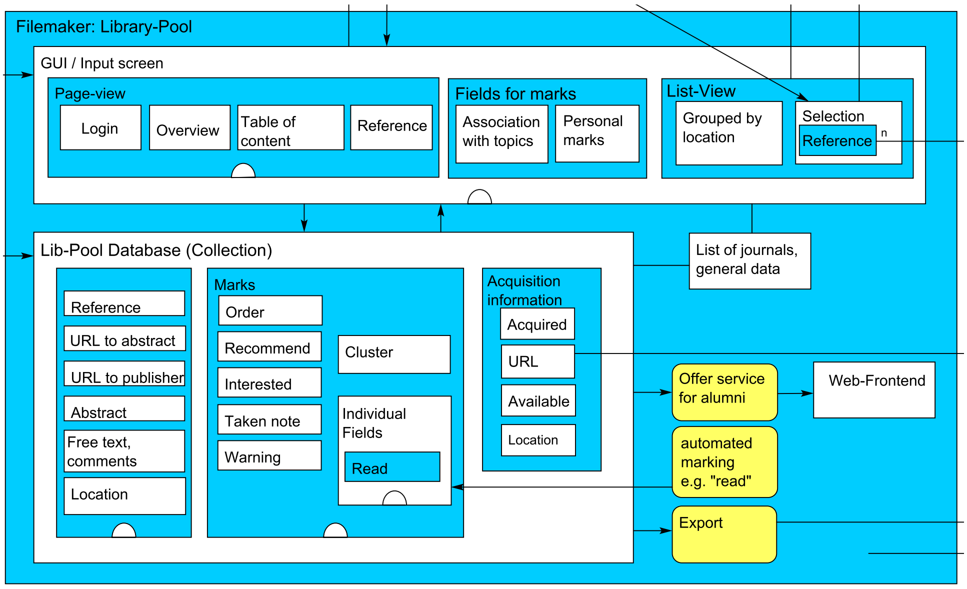


Figure 54: Socio-Technical Diagram containing a Detailed Description of the Software

Smooth Transition from Informal to Formal

With regard to the form of socio-technical self-description semi-structured diagrams have the advantage that they allow a smooth transition from informal to formal representations. The syntax of the SeeMe allows rough clustering of actions and their technical support (e.g. Figure 31) which reminds of sketching. But at the same time SeeMe includes all elements needed for detailed data modeling. Thus semi-structured diagrams seem to be especially suitable to support the representation of socio-technical systems.

Pure sketching techniques such as rich pictures [Checkland (1999)] support the early phases of a project well and allow the users to provide input very easily. But since their origin lies in organizational problem solving rather than in software-development (cf. section 4.3.3, p. 118), they miss any support for the formal representation of technical elements and the relation of work-procedures to those elements. Formal modeling techniques (use case diagrams, interaction diagrams, in the UML; Petri-Nets etc.) on the other hand concentrate very early on the final goal to unambiguously define a software-system and enforce the formally precise representation of uncertain situations.

My two case studies were needed to sharpen the idea of socio-technical self-description as a new type of documentation in software engineering projects. Part of the problem was to define its place within the established procedures and documentation of software engineering. In both case studies semi-structured diagrams using SeeMe turned out to be helpful in mediating between the formal necessities of software-development and the contingencies of human action. A next step would be to conduct a software engineering project and systematically assess the relation between socio-technical self-descriptions and other documents of the software-lifecycle (cf. chapter 9)

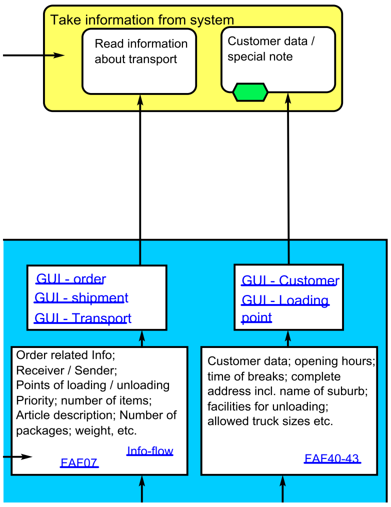

Take information from system
Read information about transport
Customer data / special note
GUI - order
GUI - shipment
GUI - Transport
GUI - Customer
GUI - Loading point
Order related Info; Receiver / Sender; Points of loading / unloading Priority; number of items; Article description; Number of packages; weight, etc.
Info-flow
EAF07
Customer data; opening hours; time of breaks; complete address incl. name of suburb; facilities for unloading; allowed truck sizes etc.
EAF40-43

Figure 55: Redundant Representation of SpiW-Com within one Diagram

Example 31: Redundant Description of SpiW-Com within one Diagram

Example 29 illustrates how the representation of the CSCW-system within the StSd changes in the course of a project. Especially within case study 1 this lead to the situation that within one diagram the same aspect of the CSCW-system was described by more than one element. Figure 55 is a detail of Figure 72 (in the appendix) and was used to discuss the GUI-prototypes within smaller meetings between STWT-2 and STWT-3. The diagram shows how the driver can use SpiW-Com to retrieve additional information about his customers and jobs (*Take information from system*). Before the prototypes existed, two entities described the information wanted by the drivers (*Order related Info: …* and *Customer data: …*). When the GUI-prototypes were available, they were linked into the diagrams (*GUI-order*, *GUI-shipment*, *GUI-transport*, *GUI-customer*, *GUI-Loading-point*). So now the information available to the driver is illustrated by the entities as well as by the GUI-prototypes. This is problematic in two respects. For one, it makes the diagrams unnecessarily crowded; second when the GUI-prototypes are discussed and new ideas are incorporated, they do not match the old description in the entities anymore, so in order to avoid contradictions within one diagram, the old entities would have to be updated. Instead, the old descriptions were removed once it had been validated that the GUI-prototypes fulfilled the documented requirements and could be used as a basis for discussion.

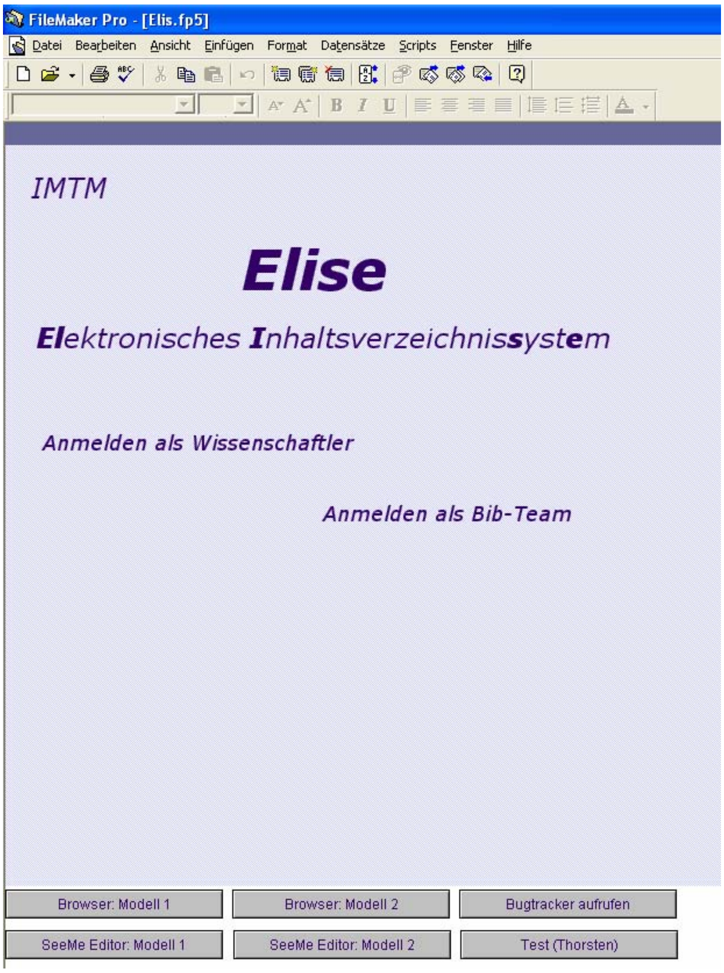
Avoiding Redundancy in the Diagrams

Example 29 and Example 31 show how topics that were modeled using the graphical notation are subsequently replaced by links to GUI prototypes.

As the semi-structured diagrams were used in case study „Mobile Communication System" there is incline in granularity from the semi-structured diagrams to the GUI-prototypes. The latter contained every field within every mask the driver has to go through to complete a certain task; the semi-structured diagrams on the other hand gave the organizational context on a level suitable for discussing the relevant topics (cf. section 7.1.1). Most of the links in the diagrams do not refer to a single mask of the GUI but to a sequence of related masks. It is important that the semi-structured diagrams do not pick up this level of detail. If they did, redundancy would again be introduced. The semi-structured diagrams should concentrate on providing additional information; if a workflow is encoded into the software system it does not need to be duplicated within the diagrams. But the diagrams should illustrate how the encoded workflow is embedded into tasks and rules that are not encoded into the software. This way the interplay between the different levels of formality shown in Figure 16 is balanced between the prototype and the semi-structured diagrams.

Example 32: CSCW-System referring to StSd

So far all examples showed links from a semi-structured diagram to a CSCW-system. In case study 2 we came to a point where mutual reference was introduced. Figure 56 shows the initial screen of ELISE which includes buttons to invoke the diagrams shown in Figure 107 to Figure 110 (in the appendix). This way the diagrams could serve as an additional resource for help to the users of ELISE.



IMTM

Elise

Elektronisches Inhaltsverzeichnissystem

Figure 56: ELISE - Initial Screen with Reference to Diagrams

When the user clicks one of the buttons referring to a diagram, the file address of the diagram is passed to the computer system's standard browser as an URL via HTTP; the browser downloads the file and invokes the SeeMe-Editor, because the file extension CME is connected to the SeeMe-Editor. The GUI illustrated in Figure 56 offers two buttons for each diagram (model). One which starts the SeeMe-Editor as described above and one which passes a graphic file (GIF, PNG or JPG) to be displayed directly by the browser. The latter was added for those users who do not have a SeeMe-Editor installed on their computer.

Mutual Reference – Design Implications for CSCW-Systems

During development, when the CSCW-system is represented by specification documents or GUI-prototypes, it is quite feasible to use the semi-structured diagrams as the main resource for documentation and to provide links to the evolving prototype. But once the CSCW-system is in use, it will be the main resource for the users; so it makes sense to include mutual references between diagrams and CSCW-system and to provide links from the CSCW-system to the diagrams as well. Example 32 shows a first attempt of this mutual reference in ELISE. From the experiences with implementing these references to self-descriptive diagrams within a CSCW-application, design implications for CSCW-systems as well as for the SeeMe-Editor can be derived.

If CSCW-systems are interpreted as part of a socio-technical setting as illustrated in Figure 6, then they are interpreted as carrying a part of the organization's self-description, namely that part that is fixed by inscription into the technical artifact. But then CSCW-systems are also part of the interplay between different forms of socio-technical self-descriptions as it is illustrated in Figure 16. And from this point of view it is desirable that CSCW-systems support links to the other two forms of self-descriptions, namely ephemeral and documented self-descriptions. The design implications derived from this observation, address the types of software systems listed in section 2.2.4.

With regard to ephemeral self-descriptions, CSCW-systems should be combined with systems for e-mail or chat that provide users with additional means for communicating about the usage of the CSCW-system. There are CSCW-systems that already provide a combination of different types of functions – Lotus Notes® is a well known example that combines document management functions with e-mail functions. But my point here is that all types of CSCW-systems – as identified in section 2.2.4 – should foresee a communication channel that allows users to engage in a meta-communication about the usage of the CSCW-system. This concept bears relations to Robinson's concept of double level language, which requires that common artifacts should support *"implicit conventionalized communication and open dialogue with indexical focus"* [Robinson (1993a), p. 200].

With regard to self-descriptive documents such as the semi-structured diagrams, CSCW-systems need to provide interfaces for attaching such documents in a way that they can be invoked by users who want to view further information about the context of use. This implies two sets of requirements: first, the CSCW-system must include an interface that allows users to add context-sensitive links at specific places in the GUI of the CSCW-system; these links would refer to the diagrams. Second, there must be an interface by which the CSCW-system can start an editor that presents the diagrams along with additional material referenced by the diagrams. It seems to be reasonable to use standard protocols such as HTTP at this point, so that the CSCW-application does not need to address one or more specific API. This however implies that the editor which presents the diagrams is capable of opening diagrams and interpreting parameters which are passed from a CSCW-application. Section 7.3.1 (pp. 234) discusses design implications for the SeeMe-Editor that were derived from the case studies.

Template for Socio-technical Self-descriptions

The creation of the semi-structured diagrams was first guided by general layout recommendations derived in other projects using SeeMe (cf. section 5.2.3, p. 146). These recommendations foresee the separation of roles, activities and entities in form of layers. In the course of case study 1 however a refinement of these guidelines with respect to the representation of the technical system evolved: the separation of the CSCW-system into its GUI on the one side and the background functions and data on the other (a typical example is illustrated in Figure 78 in the appendix). This separation is consistent with standard software engineering methods and has the advantage that the granularity in which internal functions are described is independent from the degree of detail in the description of the man-machine interaction.

In case study 2 this separation did not evolve as clearly. The diagram illustrated in Figure 96 in the appendix is one example how the separation of GUI and internal data structure was also modeled in case study 2; but other diagrams do not include this separation. The reason for this may be that ELISE hardly provides any automated functions and that the discussions with the users concentrated on the GUI.

Besides the positive experience in case study 1 and the compatibility with standard software engineering approaches there is another argument for visually separating the man-machine interactions from the background functions: it helps to illustrate the division of labor between CSCW-system and human actors. This distinction was one aspect in Eason's design of socio-technical information systems (cf. section 4.3.3, p. 112) and it can be supported this way.

The recommendations for supporting socio-technical self-descriptions include a template for organizing semi-structured diagrams which includes the insights discussed here (cf. section 8.2, Figure 60).

7.3 Facilitated Workshops

Now that contents and form of StSd have been discussed in detail, the third aspect of the methodological support shall be analyzed. The theoretical consideration, that organizational change requires communicative processes and the best practices identified in the related work lead to the decision to include facilitated workshops into the framework for StSd (cf. section 5.2.3). The organization and realization of the workshops were the major intervention into the course of the projects and appendix 11.1 includes lists of all workshops we conducted. Example 33, Example 34, and Example 35 describe three prominent workshops in more detail to illustrate how the methodological aspects described in section 5.2.3 were realized.

Example 33: Workshop for Evaluating a Prototype in Case Study 1

Following the initial phase of analysis, the software engineers in case study 1 elaborated a set GUI prototypes for the mobile devices to be used by the drivers. We conducted a STWT-workshop ("STWT-3" in July 2003) during which these GUI prototypes were subject to evaluation.

We prepared the workshop by attaching sequences of GUI screenshots to the diagrams showing the work procedures such that the mutual relation between GUI and work procedures became evident. The GUI screenshots were embedded into the context of the work procedures; and for each step in the work procedures the GUI which was designed for its support was added. Figure 77 to Figure 83 (in the appendix) show the diagrams related to this workshop. The underlined text in the entities *Pocket PC* represents hyperlinks leading to one or more screenshots of the GUI prototype. Figure 53 is a detail of Figure 78 (in the appendix) and shows the GUI that is used for the recipient's electronic signature with the related work procedure.

The participants of the workshop were dispatcher, project-leader and manager from the logistics company, a software engineer, two researchers (myself as facilitator and a colleague who joined the discussion and took minutes). There was one portable beamer available, which we used to project the diagrams as well as the prototype. In the course of the workshop we displayed and discussed the diagrams step by step; for each step where there were GUI screenshots available, these displayed on top of the diagram. Two questions were guiding the walk-through:

1) Are the screens of the GUI good to handle?

2) Do the screens and functions fit into the workprocesses?

Example 34: Qualification Workshops

In both case studies qualification workshops were conducted based on the same workshop-design ("STWT-4" in March 2004 in case study 1; "qualification workshop" in June 2004 in case study 2).

The qualification workshops consisted of two parts: they started with a training session that had the goal to enable the participants to handle the new CSCW-system. During this part functions of the CSCW-system were introduced and explained, exercises were given and scenarios were enacted so that the participants could gain practice in using the system. The diagrams illustrating the work procedures were explained and used as additional information providing a context of use for the software-functions. This first part of the qualification workshop was aimed at increasing the individual competence of each participant.

The second part that immediately followed the first was dedicated to the social level of computer supported cooperative work. The participants were asked whether there were any open issues they wanted to clarify concerning their cooperative usage of the CSCW-system (Example 18 and Example 19 provide the detailed answers to this question). From the list of open issues, the most important were identified and discussed during the workshops. Thus the second part of the workshops lead to a set of agreements concerning the computer supported cooperative work procedures.

In both case studies, future users, managers, software engineers and researchers participated. We had two beamers available such that the GUI of the CSCW-system could be projected onto one screen and the diagrams on a second one. In both case studies, diagrams taken from previous workshops were used to provide the context of use during the first part of the workshop and the basis of discussion for the second part. At the end the qualification workshops in both case studies yielded modified as well as new diagrams documenting the agreements made during the workshops (cf. appendix: Figure 81 to Figure 85 for case study 1; Figure 104 to 107 for case study 2).

Example 35: Evaluation Workshop in Case Study 2

After a period of approximately 3 months during which ELISE was used, we conducted an STWT-workshop to evaluate the CSCW-system and the related work procedures. The goal was two-fold: one, the head of the team wanted to know whether and to what extent ELISE was used by each member of the group; second, need and potential for improvement should be collected.

The participants in the so-called "evaluation workshop" (August 2005) were: 5 members of the scientific staff, the head of the team, one student worker from the library and myself. We had one beamer available which was used for projecting the diagrams that had been used in a previous workshop for introducing the new GUI of ELISE; as needed, the same screen was used for displaying screenshots linked to the diagrams or the real program for demonstrating a special feature. The discussion walked through the diagrams and was lead by two questions:

1) Did you do this and did you use ELISE?
2) Do you have suggestions for improvement?

The results were added to the diagrams mainly in form of comments (cf. Figure 107 to Figure 110 in the appendix). The discussion showed that ELISE was indeed used by all members of the team, suggestions for improvement pertained to the organizational level ("We should organize a substitution for Gabriele", "We need to agree on a date when all journals should be included", cf. Figure 107 in the appendix) as well as the technical ("Recommendation as index for sorting the tables of content", "Field for comments with information for Alex would be useful"; cf. Figure 108 in the appendix).

Example 36: Guiding Questions from Case Study 1

Table 25 gives an overview of the most prominent examples in case study „Mobile Communication System" (cf. to section 11.1 for a summary of all workshops conducted during both case studies). For each STWT workshop the question guiding the walkthrough, the diagrams that were input to the workshop and the diagrams that were elaborated during the workshop are described.

Title	Guiding Questions	Input	Output
STWT-1	Which information flows exist?	Figure 62	Figure 64
STWT-2	How could SpiW-Com support THIS activity?	Figure 64	Figure 66 - Figure 70
STWT-3	Are the screens of the GUI good to handle? Do the screens and functions fit into the workprocesses?	Figure 74 - Figure 80 without comments	Figure 74 - Figure 80 with comments added
STWT-4	Do you need additional agreements for THIS activity? Did you have all the information you needed? Did you know how to pass on information? What do you expect from the driver / dispatcher?	Diagrams very similar to Figure 74 - Figure 80 With some adaptations to the current features of the software	Figure 81 - Figure 85

Table 25: Overview of Guiding Questions in Case Study 1

The next sections discuss the experiences made with the concept of the socio-technical walkthrough (STWT), its technical support and the integration of organizational and technical issues into one workshop.

7.3.1 WS1: Electronic, Semi-Structured Diagrams as Guiding Artifacts

The first research question with regard to the workshop design asks about the experience made with the usage of electronic, semi-structured diagrams as guiding artifacts. The following sections provide further empirical details and discussions on this topic.

Example 37: Using the Diagram to get back on Topic

During STWT-1 of case study 'mobile communication system' one task was to collect information that is missing in the current work-procedures and that could be provided by a mobile communication system. A crucial point was the information available to the drivers about the recipients' sites. Problematic situations arise if the recipient's driveway is too small for the truck, if there are no adequate tools to unload the truck etc. Each driver has many stories to tell about difficult and funny situations resulting from this lack of information (the transcript in section 11.4.1 also contains examples, although they originate from a different context). After listing a few items that could serve as requirements for the software, the participants started to share stories they had witnessed. These stories had the quality of scenarios as discussed in section 7.1.2 (p. 192); they were very concrete and helped everybody imagining the difficult situations a driver could run into during his tour. Being the facilitator of this workshop, I listened for a while but then prompted the group to derive elements from these anecdotes for the diagram by saying: „We've just started collecting information, that would help the dispatcher: location, estimated time of return, queue time, current load, …" While saying this, I pointed to the activities and entities on the diagram.

Guiding Questions for STWT in Context of CSCW-Project

The initial concept for the STWT (cf. section 5.2.3, p. 145) suggested using a predefined set of questions to be answered for the diagrams. When preparing the workshops for the case studies, it turned out that this was not helpful. Literature on facilitation [Schuman (2005), pp. 10] stresses the importance (and in fact difficulty) of finding good questions for facilitation. The questions need to be selected such that the process of finding an answer supports the participants in solving the problem on which they are working. In the course of a CSCW-project there are so many different situations and so many different problems to solve that the questions need to be adapted for each individual situation.

Although the exact guiding questions need to be found for each individual project, Table 25 provides examples of such questions which may be adapted. One important characteristic of the guiding questions shall be noted. The case studies show that although a socio-technical perspective is highly desired and valued by the participants, it does not happen without additional effort. One way to ensure a perspective that integrates social and technical issues is to find guiding questions that include both aspects and can be used to refocus the group if necessary.

Walkthrough Concept in STWT

Two differences between walkthroughs that are applied as inspection techniques and the socio-technical walk-through became apparent during the case studies. First, the STWT includes creative and constructive aspects. While the cognitive walk-through or the code-walk-through are meant to validate existing artifacts, the socio-technical walkthrough also includes planning how activities could be related best to a CSCW-system. Second, the STWT is done within a group of persons while the cognitive walk-through is meant to be conducted by a single person. It is due to both of these aspects that the walk-through through the diagrams in a STWT can never be as stringent as in a cognitive walk-through. The diagrams are more a guiding artifact in the sense that they provide orientation than a script to be executed. Example 37 illustrates the way a diagram can guide a discussion without unnecessarily restricting it.

It is the facilitator's task to balance two levels of abstraction during the discussion: on the one hand the discussion needs to be open for concrete and vivid stories describing the work procedures; on the other it needs to document work procedures that are abstract enough to serve as socio-technical self-descriptions. Here the insight from scenario based design (cf. section 4.3.3, p. 120) that users need concrete scenarios to imagine a situation, is combined with the goal of providing a documented socio-technical self-description.

Example 38: Using the SeeMe-Editor for Supporting STWT-Workshops

As foreseen in the initial concept (cf. section 5.2.3, p. 146) the SeeMe-editor was used for presenting and editing the diagrams during and in-between STWT-workshops. During the meetings the advantages of the editor such as the stepwise presentation, the highlighting and very importantly the linking to other material could be employed. In-between meetings the editor was used to aesthetically improve diagrams that were only sketched during the meetings and to include new artifacts provided by the process of software-development.

The photos in Figure 57 give an impression of a typical setting for a socio-technical walkthrough. The idea that the focus of the discussion should be guided by a diagram needs to be supported by the layout of the room. One large display, clearly visible to all participants is used for displaying the main diagram that is currently under discussion. A second display is needed for presenting additional material such as prototypes of the CSCW-application.

Additional equipment such as flipcharts, pin boards, cards etc. is needed for supporting normal facilitation techniques (cf. section 5.2.3, pp. 143).

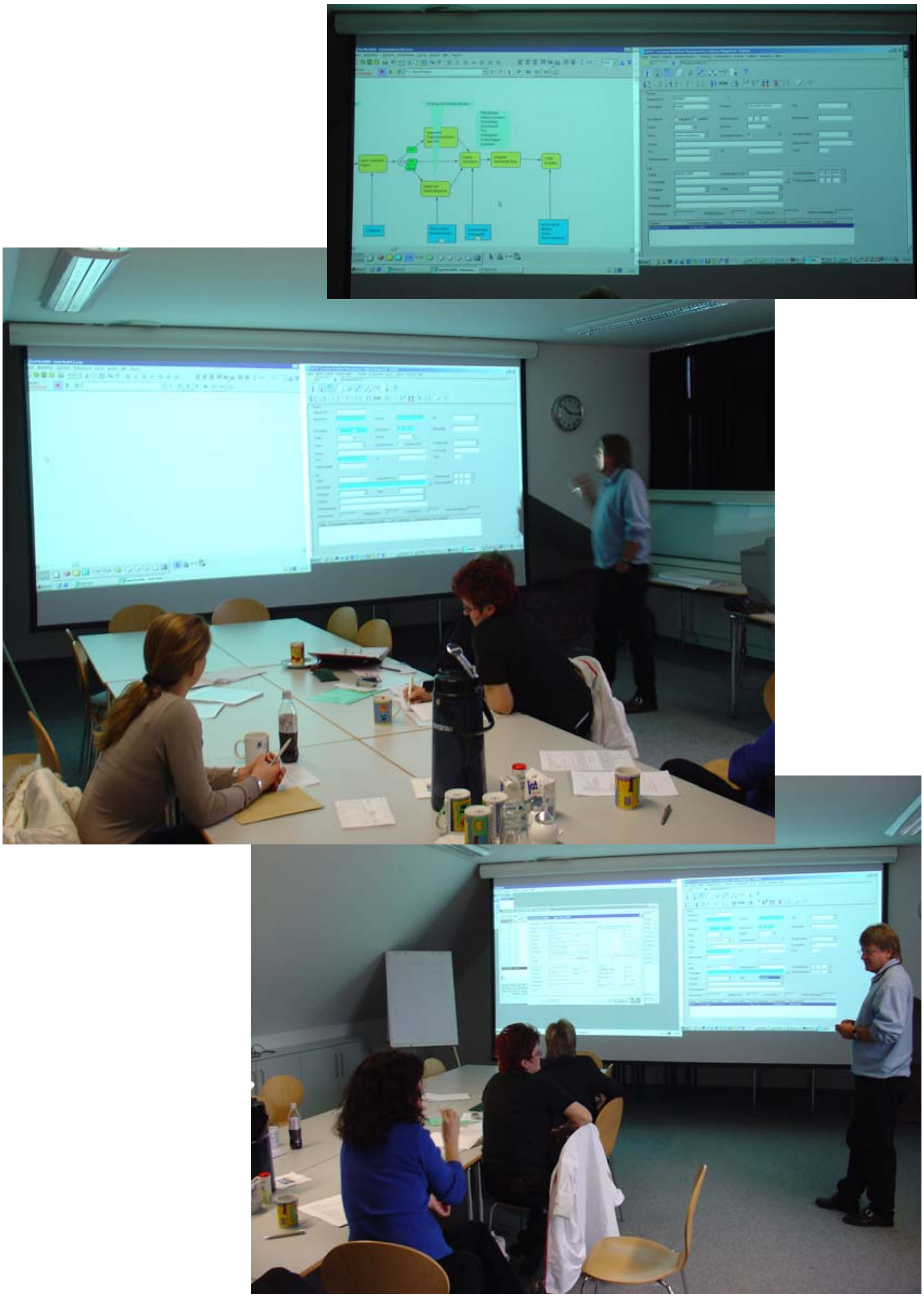


Figure 57: Workshop Setting Using Electronic Diagrams[30]

[30] The photos were taken in neither of the case studies of this thesis. They stem from another project that took place in January – April 2005 employing the methods elaborated in this thesis.

Example 39: Draftsman as a Member of the Consultancy Team

For several workshops in both case studies a draftsman supported the facilitator in the task of visualizing the results of the discussion. This was done so that the facilitator could concentrate on the task of facilitation. The metaphor for the draftsman was that he should be „the drawing arm" of the facilitator. The latter would lead the discussion, and instruct the draftsman when and how to visualize a certain aspect. Everything to do with the creation and modification of the diagrams was the draftsman's job (the transcription in the appendix, section 11.4.1 provides an example of the facilitator giving such an instruction).

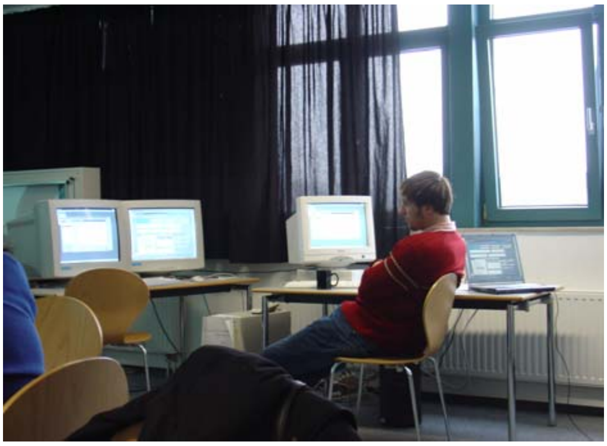


Figure 58: Draftsman sitting next to Table with Participants

Figure 58 shows the working environment of a draftsman. The scene is the same as in Figure 57. The draftsman is sitting to the right of the large table and is responsible for steering the two large displays.

Example 40: Feedback concerning Modeling during Workshops in Case Study 1

In case study „Mobile Communication System" the effort for training the participants in the modeling language was to be kept at a minimum. The participants of STWT-1 and STWT-2 were given a short introduction into the main notational elements of SeeMe. The goal was to achieve a passive knowledge; i.e. the participants should be able to understand the diagrams and request changes, it was not the goal that they should be able to create diagrams on their own.

During the qualification workshop the diagrams were drawn by a draftsman that supported the facilitator. Afterwards participants were asked whether they regarded the modeling during the workshop as helpful (cf. section 11.5.1.2 for a complete transcript of the answers). A driver as well as the manager stated that they regarded it as helpful that the results of the discussions were immediately visualized. The dispatcher however criticized that he was not able to relate the results of the discussion to the changes performed in the diagrams.

Project-Leader / Dispatcher	„I did not follow in detail, why is an arrow drawn there; I made my remarks and saw that they were included somewhere, I did not question why it was done this way … I could imagine that it is even confusing."
Manager	"Yes, of course that was a helpful support. That one had the items added directly, one could see it immediately. That was a good thing."
Driver 1	"The model was quite good … one could see what was being discussed. That was okay. Very good in fact, this way of doing it."

Table 26: Feedback concerning Modeling in Workshop of Case Study 1

Example 41: Feedback concerning Modeling during Workshops in Case Study 2

The questionnaires in case study 2 included an open question "any other remarks?". (cf. section 11.5.2, Table 38). Some participants included critical remarks concerning the handling of the diagrams during the workshops. Although the general feedback with regard to diagrams was positive (cf. Example 28), Table 27 summarizes the critical remarks for two reasons. First, they add to the dispatcher's critical feedback in Table 26 and provide useful empirical evidence for the discussion in the next section. Second, they give hints for the design of technical support (pp. 234).

Facilitator of meeting (12-April, row1)[31]	"What really bothered me was the insufficient presentation of the diagrams. For one, I was not so familiar with the SeeMe's that I would know immediately where the next step in the process, which I wanted to jump to, was located. Orientation is difficult, because the diagrams do not fit completely onto one screen, i.e. something is always cut off. The facilitator must know the diagrams really well to be able to "juggle" with them. I was not prepared well enough at this point. "
Facilitator of meeting (19-April)	"While I liked the combination of SeeMe-diagrams and ELISE in the other sessions, I felt that in this session I used ELISE more; resp. I was not able to combine the two forms as well. I think that the reason was that facilitation absorbed part of my attention."
Participant (26-April, row 2)	"I found it difficult to concentrate on the diagrams for such a long time."

Table 27: Feedback concerning Modeling in Workshops of Case Study 2

Creating and Using Diagrams causes cognitive Effort

Example 40 and Example 41 indicate that using electronic, semi-structured diagrams in workshops is not without problems. Several aspects shall be discussed in the following sections; some pertaining to the organization and facilitation of STWT-workshops, others to options for improving the technical support. This section starts by discussing the cognitive effort that is added by using semi-structured diagrams.

At the beginning of the case studies the I had expected that the SeeMe modeling notation and the SeeMe editor are so easy to understand and use that creating and modifying diagrams would not add a noticeable effort during the workshops. This expectation turned out to be wrong as the following evidence shows:

[31] The first column includes a reference to Table 38 which contains the complete set of answers.

1. I took the role of the facilitator for several workshops; and in some of them I combined the tasks of facilitating and drawing the diagrams. My experience is that it is extremely difficult to adequately concentrate on both – creating readable diagrams which correctly represent the contents of the discussion and adequately facilitating the discussion.
2. My own experience is confirmed by others who facilitated STWT-workshops; Table 27 includes two statements by a colleague.
3. Even in workshops where a draftsman was included to support the facilitator (cp. Example 39) there were breaks in which I needed to consult with the draftsman on the layout of the diagrams. During this time, the discussion among the participants continued so that I was again torn between the discussion and the modeling (the transcription in the appendix, section 11.4.1, provides an example).
4. The feedback cited in Example 40 shows that not all participants easily made the link between the discussion and the diagrams.
5. In case study „mobile communication system" the participants did not make suggestions how an issue should be drawn in the diagrams. They would make their statement and I would translate it into the diagrams. Example 1 can serve as an illustration: the driver did not say anything like „that looks so easy you should include more tasks and more branches and loops". He made his statement in terms of his work and I suggested how this could be represented in the diagram.

„Normal" facilitation techniques foresee that the facilitator is responsible for ensuring that the results of the discussion are adequately visualized for further usage. Moderation cards on pin boards are a prominent example for a visualization technique (Figure 44, Figure 45, and Figure 47 provide examples from the case studies). Usually the task of visualization is distributed between the facilitator and the participants. Sometimes the facilitator would write cards summarizing participants' statements, at other times she would ask the participants to write the cards themselves. This is possible because in general writing cards is not considered as an additional effort by workshop participants. When electronic, semi-structured diagrams are used, this procedure cannot simply be applied. Looking at the cognitive effort of modeling in detail, one can distinguish three aspects:

1. A thought needs to be translated into appropriate symbols of the modeling language. Within a semi-structured notation like SeeMe there is usually more than one way to model a situation so it also has to be decided which options will be the best for the given situation.
2. The symbols need to be drawn into a model. For this it is often necessary to rearrange elements that already exist in the diagram to make room for the new ones; or elements that are located in different regions of the diagram need to be moved closer so that they can be connected.
3. If an editor is used for the diagrams, the handling of this editor is an additional cognitive effort: the correct functions for an intended action need to be chosen and activated, and there are extra tasks such as regularly saving the file to disk or opening new diagrams.

The experience that the modeling causes extra cognitive effort which is not to be underestimated, is in line with the findings of the studies described in [Pendergast, Aytes, Lee (1999)]. The researchers found out that groups documenting a task fulfillment in form of formal models take considerably longer than those groups using sketches only. Should that lead to the conclusion that semi-structured diagrams are no suitable artifacts for CSCW-projects after all? No, the additional effort should rather be viewed as a price that has to be paid for the benefit of having a socio-technical self-description in documented form.

Three arguments for this point of view shall be given: First, the feedback described in Example 20 shows that the resulting documents are indeed valued by the participants. So, one may argue that it should be possible to motivate the participants to invest the additional effort. Second, the form that, on the one hand, causes the extra effort, is, on the other hand, what allows the diagrams to serve as boundary objects (cp. section 3.1.3, p.72). Third, there is experimental evidence that explicitly planning the usage of a CSCW-system and documenting the results in form of semi-structured diagrams leads to an increased adoption of technical features (cf. section 7.1.4, p. 202).

So, semi-structured diagrams have benefits that are worth paying the price of additional cognitive effort for their creation. During the case studies several strategies were used to alleviate the problem:

1. Delegation: the task of modeling can be delegated to a draftsman (cp. Example 39).
2. Provision of extra time: rather than working on models during discussions with all participants, the modeling can be connected to tasks that are completed in subgroups (cp. Example 42).
3. Training: the participants as well as the facilitator can increase their level of expertise such that they become more proficient in handling the diagrams.

The following sections contain detailed discussions related to these three strategies.

Pro and Con regarding the Inclusion of a Draftsman

Example 39 describes how a draftsman supported the facilitator in editing the diagrams during STWT-workshops. In a feedback immediately following the workshop STWT-1 of case study „mobile communication system”, my colleague and I as well as the student worker in the role of the draftsman unanimously agreed that the separation of facilitation and modeling was good and important. We also agreed that the division of tasks worked well. An interesting aspect was that the necessity to summarize aspects for the draftsman also helped the facilitation because it summarized the aspects for the participants as well.

A downside of delegating the modeling to a draftsman is that it further alienates the participants from the diagrams. In [Loser (2005), section 7.7.1] it is discussed that the participants' identification with the diagrams can be problematic, even without the inclusion of a draftsman. Since they have no real active part in the creation, the diagrams appear to belong to the facilitator rather than to the group. The effect is intensified when a draftsman works on the diagrams: when new contents are added to the diagrams, often parts of the diagram have to be rearranged to make room for the additions. This takes time in the range of a few seconds, up to a minute, sometimes even more. It is an awkward break if everybody remains silent until the drawing is finished; so in reality, the discussion continues. Again the facilitator is torn between discussing the diagram with the draftsman and facilitating the discussion. In response to this situation, the draftsman started to cleanup the diagrams while a discussion was in progress that did not immediately require any modeling. But this had the effect that the next time, the participants looked up to the diagrams again, they had changed and were not „theirs" anymore; a fact that adds to the problem of alienation.

Balancing the arguments for and against the participation of a draftsman, the conclusion is that a facilitator should be supported by draftsman if it can be expected that the task of facilitation is not easy. In these cases both draftsman and facilitator should be aware of the problem that they might put an additional barrier between the participants and the diagrams; however there are several options for alleviating this problem. First, the facilitator can explicate each change in the diagram e.g. by saying "The draftsman has just rearranged the activities to make room for your input". Second, the tool used for editing the diagrams should provide features for rapidly rearranging large parts of the diagram and inserting new elements. And third, there is the option of including phases during which the participants themselves work on the diagrams. An example is provided in the following section.

Example 42: Modeling during a Workshop in Small Groups

The advantage in case study „Electronic ToC Service" was that most of the participants in the workshops had an active knowledge of SeeMe, i.e. they were able to create SeeMe diagrams on their own. The facilitation concept of the qualification workshop made use of this fact. From the open issues identified in the first part of the workshop (cp. Example 19 and Figure 45) the participants selected the three most important ones and delegated their discussion to three subgroups. Each group had the task to find a solution, document it in form of a SeeMe diagram and present it to the other participants. The results are documented in Figure 104 to Figure 105 (in the appendix).

Modeling done by the Participants

One central aspect of facilitating group discussions is allowing the group to solve its own problems rather than providing them with a solution. Visualization techniques such as cardboards and flipcharts support this intention because no additional knowledge or tools are necessary on the side of the participants and everybody can take part in the discussion as well as in the visualization of results [Klebert, Schrader, Straub (1987), pp. 7]. When a tool like the SeeMe editor is used for documenting the results of a workshop in form of semi-structured diagrams, then there is an additional barrier: not only do the participants need to know the modeling language, they also need to have access to the tool. If the participants of the STWT-workshops are no more than spectators of the creation of diagrams, then an important aspect of facilitation techniques is ignored.

The need for qualifying the participants in the modeling language has already been discussed above. Here it is argued that participants also need to have access to the modeling tool in order to create diagrams by themselves. Example 42 describes how this can be organized within an STWT-workshop. The concept of delegating the solution of problems into subgroups is a standard means of facilitation (cp. section 5.2.3, pp. 143). But with regard to using electronic diagrams in workshops, this approach has two specific advantages: First, the relative additional effort of creating semi-structured diagrams is reduced. It is a standard procedure in facilitation that subgroups, who elaborate a topic, visualize their answers and present them to the rest of the group [Klebert, Schrader, Straub (1987), p. 142]. Thus, the task of creating diagrams blends in nicely with standard facilitation techniques. Second, the character of the diagrams as boundary objects becomes immediately evident. The subgroup needs a form of documentation that is robust enough to be presented to and understood by the whole group. Using semi-structured diagrams for this lets the participants directly experience their benefits.

Three requirements must be fulfilled in order to realize such a concept:

1. The majority of participants must be trained in the modeling language. They must have enough expertise to translate socio-technical issues into notational elements.
2. Several of the participants must be able to operate the editor for creating diagrams.
3. The technical equipment must be available such that several groups can use a diagrammatic editor at the same time.

All three aspects are discussed in the next sections.

Expertise in the Modeling Notation

The cognitive overhead induced by the usage of a semi-structured modeling notation can be reduced for the workshops if facilitators as well as participants have enough expertise in modeling to do so with ease. The advantages of an intensive training are obvious:

- the confusion noted in the feedback of Example 40 could be avoided;
- facilitation concepts such as modeling in subgroups are made possible;
- the problems misinterpreting details in the diagrams (cp. Example 25) are reduced;
- the alienation of participants and diagrams is reduced.

However there are disadvantages to providing intensive training in the modeling notation; and in case study „mobile communication system" these disadvantages lead to the decision to conduct a minimal training only. The reasons discussed in the case study were:

1. Management was afraid that the participants (drivers more than dispatchers) would be discouraged if they had to learn a modeling notation before starting the real work.
2. Management was afraid that the workshops would take too much time (and thereby cost too much money) if the training sessions for the modeling notation were included.
3. The fact that the company already used a modeling language for their business process modeling (cp. Example 26) decreased the willingness to engage in a new modeling notation.

The conclusion drawn from this is that a certain level of expertise in the modeling notation is required if semi-structured diagrams are to be employed in CSCW-projects. This „certain level" can be specified as follows: participants should have an active knowledge that allows them to translate simple issues into diagrams; facilitators should be experts who know all possibilities of the modeling notation have experience from several projects and can trade off different options for modeling the same issue in different situations. At the beginning of the project the additional effort needs to be motivated by the benefits of having semi-structured diagrams available in the course of the project.

Expertise in Diagram-Editor

What was said regarding the modeling notation, also applies to the handling of an editor: it must be ensured that there are members of the project team who can work with the editor. Obviously, a draftsman needs to have the highest skill in handling the editor, but it is also advisable that the facilitator has excellent skill in handling it. Since the facilitator is responsible for organizing the series of STWT-workshops and since the diagrams have such a prominent role in these workshops, the facilitator should be able to edit them on his own.

In case study 1, facilitator and draftsman were indeed the only ones using the editor. However, this limits the options for workshop design as well as for organizing the tasks to be done in-between workshops. The delegation of modeling tasks into subgroups during a workshop (cf. Example 42) is only possible if there are enough participants to do the modeling; and if the task of redoing the diagrams in-between workshops should be distributed among the several members of the project team, they as well need to be capable of handling the editor.

Tool Support - Four Settings

So far, organizational issues such as delegating the task of modeling or qualifying participants have been discussed. This section turns to technical options for improving the usage of semi-structured diagrams in STWT-workshops.

	Collaborative	**Individual**
Editing	**Co-located Collaboration:** During STWT-workshops the participants collaboratively document their computer supported work procedures in form of semi-structured diagrams.	**Individual Editing:** In-between STWT-workshops the facilitator or a draftsman is responsible for aesthetically improving the diagrams and for integrating new artifacts provided by the process of software-development.
Reading	**Presentation:** Diagrams are presented to different groups of stakeholders in order to inform them about the future computer supported work procedures.	**Individual Reading:** A user informs himself / herself about computer supported work procedures by reading the diagrams; the diagrams may be included in an electronic document or as help-resources in a CSCW-application.

Table 28: Four Settings needing Tool Support

From the usage of the editor in the case studies four settings which need technical support can be deducted. Table 28 explains the four settings by describing typical situation in which they occur.

At the beginning of the case studies, the SeeMe-Editor offered the features summarized in section 5.2.3 (pp. 146); I used the case studies to derive further requirements, some of which have already been implemented[32]. The following sections discuss the tool support I suggest for each of the four settings.

Tool Support for Co-located Collaboration

For supporting collaborative editing in facilitated workshop settings, features of synchronous joint editing can be used. For the SeeMe-editor we designed and implemented a prototypical version of such features [Turnwald (2002)]. Since the methodological framework for CSCW-projects foresees facilitated workshops, those versions of joint editing are adequate that support the role of a facilitator who hands over control to the participants by turn taking mechanisms. Before joint editing features can be used in STWT-workshops, the setting described in Example 38 (Figure 57) needs to be extended such that all participants have a computer available on which a version of the SeeMe-editor is active. But when everybody has his / her own screen to look at and to edit the diagrams, attention is lost for the communicative process within the workshop. Preferable are technical solutions that support the collaborative handling of electronic media e.g. in form of large interactive displays or interactive tables. Within CSCW the term "co-located" [33] cooperation is used for such settings and research is done for designing adequate technical support ([Streitz et al. (2003)], [Rogers, Lindley (2004)]). For STWT-workshops it is desirable to have multiple, large interactive screens on which the diagrams as well as additional material such as other documents, prototypes or applications can be presented. If STWT-workshops could be conducted in such an environment, some of the problems reported by participants could be alleviated: problems concerning the visibility of diagrams (cf. Table 37, 26-April); problems concerning the fact that only small parts of large diagrams can be displayed (cf. Table 27); problems resulting from the fact that only one beamer / screen is available and that projections of diagrams have to share a screen with projections of prototypes or applications (cf. Example 33, Example 34, Example 35). An editor should support co-located collaboration by supporting large interactive displays and the features they provide in addition to the conventional settings that were used in the case studies.

But in addition to supporting better access to the electronic diagrams, an editor should include special features for supporting those tasks that lead to the additional effort during workshops. General support is needed for rapidly drawing and labeling elements and for rearranging an existing diagram e.g. by moving elements aside to make room for new ones. The SeeMe-editor includes a prototypical version of a so-called broom that provided such support [Schmitz-Becker (2002)].

[32] The implementation of the SeeMe-Editor is a long term project in our work group. The group is lead by three researchers including me; the actual coding is done by student workers [Webpage SeeMe Editor].

[33] The last CSCW conference included a workshop on the topic: [Webpage CSCW-Workshop 2004].

A feature that is standard for text editors but not yet for drawing tools is the markup of changes. When a diagram is discussed and edited during a workshop it is important that the participants can see what has been there from the beginning and what has been added, deleted or modified in the course of their discussion.

Tool Support for Individual Editing

Diagrams that are created or edited during STWT-workshops are not optimized with regard to their layout. During the workshops, it is important that decisions are made and documented; the focus does not lie on creating “nice” diagrams. This task is left to the facilitator or other members of the project team to be done in-between workshops. Loser refers to this task as “aesthetical improvement” of the diagrams [Loser (2005), pp. 189]. The tool support needed here includes standard features of graphical editors such as alignment or adjustment of element size. But additional support such as a critiquing system informing the user about deviations from the suggested style guide would also be helpful [Wacker (2000)].

It turned out in the case studies that a large part of the editing in-between workshops was caused by new artifacts that came from the software-developers. In case study 1 the links contained in the diagrams needed to be updated for each new screenshot, because filenames and locations changed. Two requirements for tool support are derived from this experience. First, there needs to be a very fast and easy way for creating and editing a large number of hyperlinks in the diagrams. Second, an integration of diagrams and additional media into one repository that supports a unified name space for resources would be helpful. Then links in the diagrams could address a resource such a specification document and the link could remain the same even though new versions of the document are provided. Another feature such a repository should support, is to keep the consistency of links when a diagram and the linked resources are copied from one system to another. This is necessary when the facilitator prepares a diagram on his local system and then transfers it to a server for making it available to other participants. In this case it should be possible to transfer the whole package of diagrams and attached material without having to change all links. In order to be able to handle diagrams outside the editor, a format that can be interpreted by other applications is needed; for the SeeMe-editor functions are developed that support XML as a storage format [Langer (2005)].

Another task to be supported is the preparation of presentations and collaborative editing sessions. One of the key aspects of the walk-through concept and of presenting complex diagrams is that a diagram should not be presented en bloc. It should rather be opened up for the participants step by step such that they can grasp its contents (cf. section 5.2.3, p. 146). Such a presentation has to be prepared by the facilitator and the case studies showed that the effort for doing so is tremendous even though the SeeMe-editor provides hide / show and snapshot functions. A tool supporting the facilitation of STWT-workshops should be able to generate default presentations for diagrams and it should provide functions for rapidly generating presentations. The SeeMe-editor includes two strands of functionality that point into this direction: First, the so-called message-concept [Turnwald (2002)]. All drawing actions performed by the user are recorded in form of messages. Originally this concept was implemented for supporting distributed joint editing; but as a side effect, an editor and player for these messages was implemented. It allows the user to bundle drawing steps and to easily replay them. The disadvantage is that the user is not completely free in rearranging the messages; she is limited to the order in which the diagram was drawn. But this concept could be further developed for supporting the preparation of presentations in general. Second, we implemented a reader [Berning (2004)] that uses heuristics for automatically presenting SeeMe-diagram. This feature is meant to support the individual reading of diagrams, but it contains elements, such as the heuristics that could also be used for supporting a semi-automated preparation of presentations.

So far the SeeMe-editor does not distinguish between individual and collaborative editing. The experience in the case studies indicates that the tasks to be performed are so different that it may be justified to provide different user interfaces and different functions for the two settings. The GUI for collaborative editing would focus on features for easy and rapid drawing of diagrams; it would provide support for users who are not familiar with all details of the modeling language or the handling of the editor. The GUI for individual editing would address a more experienced user and would provide functions for optimizing the layout, integrating other media, and preparing presentations.

Tool Support for Presentations

When socio-technical self-descriptions are employed in the context of CSCW-projects there will be many situations in which diagrams need to be presented. The three most typical situations are:

1. The result of the last workshop is presented at the beginning of an STWT.
2. Diagrams are presented in a step by step manner to allow "walkthrough" during an STWT-workshop.
3. Diagrams resulting from an STWT-workshop are presented to a different group of people for informing them about the project.

It has been mentioned that semi-structured diagrams that serve as socio-technical self-descriptions need to be embedded into a communicative context (cf. section 4.3.5), the tool support for presenting diagrams should help the presenter to create such a communicative context by increasing the readability of the diagram and by encouraging communication. The SeeMe-editor includes features for these purposes. The presenter can freely prepare the presentation of the diagram by predefining so-called snapshots which she can tailor according to the flow of her presentation. Stepwise display of the diagram increases the readability for the audience. Problems occur when one wants to switch between presentations with the editor and other programs such as Microsoft's PowerPoint®. Here a better support for smoother transitions would be helpful.

A new feature that is being implemented and tested adjusts the overall size of the diagram when elements are hidden from view [Schmitz-Becker (2002)]; this alleviates the problem of panning large diagrams on a small screen.

Tool Support for Individual Reading

Although much emphasis is put on the fact that semi-structured diagrams that serve as (socio-technical) self-descriptions need to be embedded into a communicative context, there are situations in which a single person needs to read the diagrams. This can be the case in-between STWT-workshops if participants read diagrams from the workshop; but it can also be the case when the CSCW-system is operational and the diagrams are used as an additional resource for information. The technical support for this situation should ease reading and understanding of the diagram.

For the group situation, it was said above that features such as the incremental display of elements support the readability. These features can be transferred to the setting in which an individual reads the diagrams. For the SeeMe-editor we designed and implemented a semi-automated presenter for this purpose [Berning (2004)]; it presents diagrams either completely automatically or steered by the user who triggers each step of display. There are two ways to determine the order in which the elements of the diagram are presented. First, the SeeMe-presenter contains syntax-based rules from which it automatically generates presentations for SeeMe-diagrams. Second, the editor of a diagram can adjust the rules such that he influences the way the diagram will be presented to users. These adjustments can vary from slight modifications of the default up to a complete rearrangement of the pre-configured presentations.

A problem is that the user needs to open a special editor to view the semi-structured diagrams; and within the editor he has to select a complete diagram and search for the parts he is interested in. It is discussed above the integration of diagrams and other material such as screenshots and photos within one repository would be helpful. For the situation of a single user reading diagrams, the technical integration of the diagrams with the CSCW-application itself would be helpful. It should be possible to open up diagrams for additional information from appropriate places within the CSCW-application (cf. Example 32). There are several options for implementing such integration. The first option is to export the diagrams into standard graphic formats (GIF, JPEG, PNG). The advantage is that it is usually no problem to link them into a CSCW-application; programs for displaying the graphics are included in the operating system e.g. in form of browsers. There are two major disadvantages to this solution: First, each time the diagrams change, the exports have to be regenerated integrated. Second, special features such as linking additional media or the automated presentation of the diagrams are not available. In case study 1 the SVG format was used for export which supports hyperlinks, so at least the additional media could be integrated.

An approach that is much more favorable is to integrate the editor itself with the CSCW-application. The editor needs to provide an interface such that it can be started from the CSCW-application, and the latter needs to provide functions for doing so. An important requirement is that the editor is not only started but that the features for presenting diagrams can be initiated. It should be possible to start an automated presentation of a certain diagram or to display a specific snapshot. We are currently investigating two directions for implementation. The first option is using SUN's JAVA™ Network Launching Protocol (JNLP) [Webpage SUN, Java Web Start], which allows installing and invoking JAVA applications throughout a network. The second option is using a new dedicated file type describing a SeeMe-diagram as a resource including information about a specific snapshot to be displayed, the mode of the reader to be started, the location where additionally linked material can be found etc. In either case, the CSCW-application would need to allow the placement of context-sensitive hyperlinks which point to resources that can be opened by a standard browser.

In connection with the individual reading of diagrams that are available as a shared resource, another setting which is not included in Table 28 could become relevant: distributed, asynchronous joint editing. When the diagrams are accessible from the

CSCW-application, users may want to start to edit them to mark changes in procedures or to indicate areas that need further discussion. For this again, it is necessary, that the diagrams are available in a repository that is reachable by all users.

No Tool Support for Distributed, Synchronous Joint Editing

Why is the setting of collaborative editing limited to co-located collaborative editing (cf. Table 28)? One could also think of situations in which two or more persons from the project who are not collocated would like to work on a diagram together. This would call for features supporting synchronous joint editing in distributed settings. Indeed, the SeeMe editor supports such features in a prototype version [Turnwald (2002)], but neither of the case studies required this support. The reason for this lies most likely in the workshops that are foreseen within the framework for StSd. The participants meet in regular intervals for discussing and editing the diagrams, so the need for additional editing sessions in-between is reduced. This is not to say that there will not be cases in which such features would be very helpful, but they are not central to the support of socio-technical self-descriptions in CSCW-projects.

7.3.2 WS2: Integrating Technical and Organizational Issues in one Workshop

The previous section discussed the facilitation techniques for STWT-workshops; this section turns to the experience made with combining the discussion of technical and organizational issues in one workshop. Many of the examples discussed so far showed how discussions of organizational and technical issues were integrated with STWT-workshops. Example 33, Example 34, and Example 35 serve as concise summaries of prominent workshops conducted in the course of the case studies. So the general statement is that the integration of organizational and technical issues within one workshop setting was possible; however there are two examples that shall be given from which lessons can be learned for conducting such workshops.

Example 43: Software engineer making Promises about Usage of CSCW-System

One of the big issues in STWT-2 of case study 'Mobile Communication System' was whether the new system is suitable for checking on the drivers. The technical issue that initiated the discussion once more was whether or not a field is needed in the GUI into which the driver enters his reasons for choosing a certain route (cp. also Example 9). During the discussion one of the drivers expressed his concern that the dispatchers would regularly check the drivers' positions on the route. In case of necessary detours the dispatcher could suspect the driver of not doing his best to reach the customer on the shortest possible way. Both project leader „Westkreis" and one of the dispatchers assured the driver that this would not be the case. The participating software engineer then added (Translated from transcript in section 11.4.4):

> *„This will not be the case! I can assure you as well that this will not be the case. The system can just be used to ask. It is not a controlling system, the data are recorded, somebody will have a look at them, and once in a while one would ask and then the person explains I drove a detour, the road was closed. That's it. The system is applicable*

> *only when things occur repeatedly … The data are just recorded to control the progress toward the completion of the job on the one hand, and to derive insights for the planning for the future on the other. Not to check on the road you are just driving."*

Leave the Problem with the Organization

Example 43 illustrates how a software engineer tries to attach organizational procedures to the technical characteristics of a CSCW-system. He explains that SpiW-Com does provide the means to control the drivers more closely, but he assures the drivers that these features will not be used against them. Undoubtedly the software engineer has good intentions, but he promises things he simply should not promise because it is beyond his abilities to ensure them. Example 17 that describes the initial discussion about the future use of mobile phones includes a similar sequence. The participating software engineer ensures the drivers that they will still be allowed to use their mobile phones once SpiW-Com is operational. The development of the discussion later on in the project relativized this statement (cf. Example 14).

These examples show how the levels of the two processes within a CSCW-project (cf. Figure 11) can be confused. It is an issue of requirements-elicitation to find out what data are needed to be processed by SpiW-Com and it is a matter of software engineering to design the technical processing of the data; but it is a matter of structural coupling to decide about the organizational processing of the data. The question how data available with SpiW-Com will be used within „Westkreis" is clearly an issue to be debated and agreed upon within the organization. Once the system is operational, the software engineer will not be available anymore and his ideas about the usage of data will be irrelevant. Agreements among the members of the organization are needed, because they are integrated into the communicative network that makes up the organization and they can be referred to in case of conflict.

With reference to the concepts of systemic intervention (cf. section 4.2) one could cite the general principle „Leave the problem with the social system"; neither the facilitator nor the software engineer can decide how the data will be used, so they will have to have the patience to allow the members of the organization to come to a conclusion.

The facilitator and the participants of workshops working on StSd need to be aware of the two processes that are combined within a CSCW-project. And especially the software engineers need to be aware of their limits. They can design the technical CSCW-system, but they can only influence and never determine the structural coupling of the organization with the CSCW-system. It is the task of the facilitator to recognize and correct situations like the one described in Example 43.

Example 44: Putting aside Organizational Issues

During STWT-2 in case study „mobile communication system" the issue was raised how new information that the drivers enter during their tour should be displayed on the dispatchers' screens. One option was that the information should automatically be displayed by means of pop-up windows, beeps, a small icon etc. ("push mechanism"); the alternative was that the dispatcher would have to trigger each display ("pull mechanism"). During this discussion the participating software engineer explained that he planned to provide a configurable mechanism; the users should be able to determine how a certain class of information would be handled. He said (translated from the transcript in section 11.4.2):

> *„… we want to foresee the possibility of prioritizing certain data. So, when THIS happens, then I – the dispatcher – will be informed immediately; other things will be put into an intermediate storage and I need to retrieve them. The name for this is push-and-pull information. … That will be prioritized, … the data is available and at some point a competent person needs to tell us this, this this and this do this way and this, this, this and this do that way … the option is conceptually prepared."*

From the software engineer's perspective the debate could have been ended at this point. However the debate continued with the issue who would benefit from the new way of information processing. After a while of discussion, the facilitator came back to the original question and the participating users concluded that they preferred a pull-mechanism.

Giving Rights to Organizational and Technical Processes within one Workshop

With regard to the process model for CSCW-projects (cf. Figure 11) Example 44 describes a workshop that comprises discussions for both processes – organizational change as well as technical design. With regard to the process *Supporting organizational change / structural coupling*, this workshop was the first in which drivers and dispatchers discussed future, computer supported work procedures. In this respect the discussion in the workshop was the first step towards a structural coupling or a congruent technological frame in the sense of Orlikowski (cf. p. 69). With regard to the process of *System Design* the workshop was part of the requirements analysis; there were still open questions regarding the system design, one of them was the question about push- and pull-mechanisms.

At some point of discussion, it was evident for the software engineer that the software needed to implement different mechanisms for passing information; it also seemed to be reasonable to realize a flexible implementation that includes classification of the information by means of configuration. The software engineer had all input he needed to do his job at this point, and implicitly suggested closing the debate by stating that at some point „a competent person" should inform him what the initial configuration should be.

The idea of making the classification configurable is a good one that supports the structural coupling in the early phases of usage, when experience will bring up needs for change. However this passage conveys problematic views of some software engineers – and maybe computer science in general – on organizational processes. The software engineer in Example 44 implies that the organization's decision about the classification of information will suddenly be there at some point in time, conveyed by some "competent person". He does not at all relate to the social activities that are necessary for reaching this decision.

What are the lessons that can be learned from this example? First, within the workshops it is the task of the facilitator to make sure that the needs of both processes are taken into account. But to do so, she has to keep in mind that there are two processes and that they have different needs. For this it is helpful that an overall process-model including the two processes is known to all participants. Since such a model was developed during and after the case studies, it had not been available at their beginning. Especially in case study 1 where the participants were not integrated into a research context of „socio-technical" issues, this turned out to be a disadvantage. Second, participating software engineers take a dual role within a workshop like the one in this example. With regard to the process of *System Design* they are a member of the construction team and they need to gather information from the future users for implementing the software. With regard to the process of *Supporting organizational change / structural coupling* they are a consultant with technical expertise; but as a consultant they need to respect and provide room for the process of structural coupling which has to be performed by the users.

7.4 Integrating SWE and Organizational Change

So far the methodological concepts of socio-technical self-descriptions, their content, their form and their integration into the communicative setting of STWT-workshops have been discussed. This section is dedicated to the integration of the two processes of organizational change and software design on project level (cf. Figure 11). For this, two questions are discussed: the first asks whether socio-technical self-descriptions could serve as boundary objects between the heterogeneous groups involved in a CSCW-project. The second asks whether the methodological effort was successful by asking about the outcome of the projects.

7.4.1 I1: Socio-Technical Self-Descriptions as Boundary Objects

Were StSd used as boundary objects?

Research question I1 asks whether socio-technical self-descriptions in form of semi-structured diagrams could serve as boundary objects throughout a CSCW-project. To answer this question I check whether the diagrams created during the projects can be considered as boundary objects by the standards of Star's definition cited in section 3.1.3 (p. 72). The examples provided from the case studies so far serve as empirical evidence.

Boundary objects are meant to support the cooperation of heterogeneous groups whose members have different viewpoints on the same subject. Both case studies include the cooperation of heterogeneous groups such as users, management, software developers, and researchers; so it is reasonable to look for objects that cross the boundaries between these groups. An interesting aspect is that the user groups themselves were not homogeneous: in case study 1 "mobile communication system" the group of users comprises the drivers who would use SpiW-Com on the mobile device and the dispatchers who would use a different version of SpiW-Com on their office computers. In case study 2 "electronic ToC-service" the group of users comprises the scientific staff who reads the table of contents and selects the articles; and the student workers in the library who check the selections and retrieve the desired articles.

In both case studies diagrams were created that do indeed include information relevant for different stakeholders and different strands of discussions within the projects; three examples shall be reconsidered to argue that the diagrams did function as boundary objects.

First, the diagram of Figure 27 in Example 1 (which is a detail of Figure 64 in the appendix) functioned as boundary objects between drivers and dispatchers in case study 1. Both the work procedures of drivers and dispatchers are illustrated; they are loosely coupled by the information that is exchanged. Thus it was possible (as it is described in Example 1) that the drivers could add information that was relevant for them quite independently from the dispatchers without disturbing the other information contained in the diagram. A second example is Figure 36 that integrates information from different strands of action within case study 1. On the one hand it describes the way drivers and dispatchers agreed to cooperate when planning and transferring a new tour for the driver (cf. Example 10 for a more detailed description); on the other hand it describes the usage of the mobile client by connecting the activities with entities containing links to screenshots of the prototype. While the first type of information is needed by the drivers and dispatchers for organizing their cooperative work, the latter type provided the context of use for the group that validated the prototype. The third example refers to Figure 54 (which is a detail of Figure 96 in the appendix) taken from case study 2. This diagram includes workflow information relevant for the future users of the system and detailed data descriptions relevant for the database designer. During the course of the project, the workflow information was refined and enriched by conventions guiding the cooperation while the data descriptions were replaced by links to the evolving prototype. Figure 107 to Figure 110 (in the appendix) show the according diagrams at the end of case study 2.

The three examples show that the semi-structured diagrams created during the projects were indeed able to adapt to the local needs of different stakeholders. At the same time they provided a valid overview of the whole socio-technical setting which was successfully employed during the qualification workshops (cf. Example 34; Example 20 and Example 22). Thus one can conclude that socio-technical self-descriptions in form of semi-structured diagrams can function as boundary objects between heterogeneous groups within a CSCW-project.

The discussion above concentrated on the content of the diagrams for arguing whether or not they functioned as boundary objects. Another indicator is their usage in quite different situations. The following section discusses how the diagrams worked as boundary objects by being guiding artifacts within and between different phases in the projects.

StSd as Guiding Artifacts through different Phases

In both case studies, semi-structured diagrams were used from the very beginning to the very end for documenting the organizations' socio-technical self-descriptions. Representative overviews of diagrams from all phases are provided in the appendix in sections 11.3.1 and 11.3.2; Table 33 and Table 34 relate the diagrams to the phases in which they were created. It is important to note that in both case studies, there was a continuous use and development of the diagrams. The diagrams were taken from one phase to the next and adapted as the software-development and the agreements on work procedures advanced. The metamorphosis of the daily record in case study 2 is a very illustrative example which is detailed in Example 29.

The idea of having artifacts that provide a thread through the whole project has been identified as a good practice in earlier sections: methods of software engineering suggest use cases for this purpose (cf. section 4.1.5, p. 99); scenario based design creates scenario descriptions for this purpose (cf. section 4.3.3, p. 120). The question was whether socio-technical self-descriptions could be used as guiding artifacts to keep a socio-technical perspective throughout the project. The experience documented by the usage of StSd in both case studies is positive.

In case study 2 the diagrams were also transferred from the design into the use phase: STWT-workshops continued after ELISE had been deployed (cf. section 11.1.3; workshops in May, June and August 2005). Within these workshops the users exchanged experiences and advice, brought up new requirements, or discussed and modified conventions. At this point, the concepts of appropriation (cf. section 3.2.1, p. 83) and socio-technical self-descriptions meet. Socio-technical self-descriptions are not restricted to a design phase prior to use, once the CSCW-system is operational and the organization appropriates it, they continue to be helpful by explicating the organizational context of use and thus making it available for reflection and change.

In spite of the overall positive experience with socio-technical self-descriptions as boundary objects and guiding artifacts, there were also some problems with regard to the content and form of the diagrams; the following sections contain empirical data and a discussion.

Example 45: Confusion about Purpose of StSd in Case Study 2

In case study „Mobile Communication" the early diagrams describing the drivers' and dispatchers' work-procedures (e.g. Figure 62, Figure 64 in the appendix) caused some irritation among the partners in the research team. The purpose and therefore the scope of the diagrams were not obvious to the software engineers on the one hand and the logisticians on the other. The software engineers had two main expectations: first, they expected diagrams that would provide them with clear cut requirements for the software. Second, they needed workflow descriptions that could be directly transferred into logic for software-components. The logisticians were familiar with diagrams on the level of business processes that they could use for economic considerations such as calculating costs for certain transactions. Socio-technical diagrams are similar enough to other existing diagrammatic methods to give rise to expectations in both directions which subsequently they do not necessarily fulfill. The other researchers in the team were quite uncertain in their evaluation of the socio-technical diagrams. In the beginning they could not understand the reasoning for the inclusion or exclusion of information and came to the conclusion „those diagrams do not help us". Later, after the qualification workshop of case study 1, the participating software engineer was asked whether the semi-structured diagrams were helpful in the workshop. In his answer he weighed the pro's and con's (cf. section 11.5.1.2, p. 368):

> *"A picture says more than 1000 words. Therefore the modeling is helpful. … For each method there are advantages and disadvantages. … The advantage* [for SeeMe] *is that the notation can describe weakly structured work procedures - that is obvious. But that induces the disadvantage: certain analyses that are based on certain formalisms cannot be done."*

Example 46: Workflow-Control derived from StSd

The architecture of SpiW-Com in case study 1 includes a workflow component by which the functionality of the mobile client is controlled (cf. section 6.2.2, Figure 21). Such a workflow component comprises workflow engines for executing workflows which are represented by workflow descriptions. Since the workflow descriptions used by the workflow enactment service of a workflow management system are needed in a formal syntax which does not lend itself well to be discussed among users and designers, separate, mostly graphical, process descriptions are used for this purpose. These graphical representations are then transformed into workflow descriptions that can be interpreted by workflow engines. [Haase (2003)] discusses these issues for SpiW-Com. He comes to the conclusion that SeeMe-diagrams are suitable for graphically representing workflows; the graphical representation is exported in XML-format which can then be used by the workflow component. But in order to use SeeMe-diagrams as an input into a workflow management system, they must adhere to strict rules which are also listed in [Haase (2003), pp. 59]. These rules include:

- activities may only be connected to entities
- entities may only be connected to activities
- all element names must be unique

In order to gain empirical insight into the generation of workflows, we generated some SeeMe diagrams which adhered to the standards needed. However we did not pay attention to the necessities of the workflow engine when creating and discussion self-descriptive diagrams in the workshops.

Boundary Object but no Catchall

Part of the initial irritation described in Example 45 was due to the explorative character of the case study. At its beginning it was guided by the rough idea to represent work-procedures and their technical support; no clearly defined concept existed that could be used for explaining the idea of StSd thoroughly to the other partners in the project team. An important lesson learned from this case study is that in any new project the framework of StSd (cf. section 8.2) needs to be introduced in the very beginning of the project so that everybody knows what to expect from the diagrams. This worked better in case study 2 where problems like the ones described in Example 45 did not occur.

But Example 45 in connection with Example 46 can be used to discuss where the idea of StSd as boundary objects reaches its limitations. The main purpose of semi-structured diagrams as socio-technical self-description is supporting the evolvement of a socio-technical organization (cf. Term 20) in which a CSCW-system is integrated into the organization's work procedures. The case studies provide empirical data regarding the content (cf. section 7.1, Table 18) and form (cf. section 7.2) of socio-technical self-descriptions. StSd cannot serve as boundary objects anymore when this implies compromising characteristics of StSd that have been identified as important in the case studies.

The process descriptions explained in Example 46 are one example where the limits are reached. These were needed for implementing the workflow engine and thus the requirements regarding their form and content could not be reconciled with the needs of socio-technical self-descriptions. Regarding the form, the sequence of states and transitions required for the Petri-nets is not in line with the drivers' and dispatchers' way of describing their work procedures. It is discussed in section 7.3.1 (p. 229) that even the creation of semi-structured diagrams means an additional overhead in comparison to the creation of texts or freehand sketches. This overhead was regarded acceptable because the resulting documentation was valued. If we had observed the rules cited in Example 46 the corset of the modeling language would have been even tighter and their creation in the context of participatory workshops would be even more difficult. It is an interesting question for further research how the two diverging needs of workflow descriptions for implementation and socio-technical self-descriptions for the evolvement of a socio-technical organization can be reconciled. One could think of semi-automatically extracting a workflow description suitable for implementation from the richer socio-technical self-description.

The goal of both of my case studies was to identify the essential characteristics of documents that may serve as socio-technical self-description in the context of CSCW-projects. The experience from the case studies shows on the one hand that such documents can serve as boundary objects between users and software engineers; but on the other hand the experience also shows that there are limitations when the function of being a boundary object would be at the expense of characteristics essential for the function of socio-technical self-descriptions. It is an interesting aspect for future projects to investigate in how far more documents and models needed for software engineering can be (automatically) derived from socio-technical self-descriptions without compromising their essential characteristics.

7.4.2 I2: Socio-Technical Setting as Outcome of CSCW-Projects

The last research question to be answered by the analysis of the case studies concerns the outcome of the projects. Was it indeed possible to support the evolvement of socio-technical settings as defined in section 3.2.1? Two further examples from the case studies shall provide the empirical material for discussing this question.

Example 47: ELISE is used

The CSCW-system ELISE in case study 2 is fully operational since May 2005: a group of 8 scientists and 1 student worker in the library is using ELISE on a regular basis for handling the scientific periodicals subscribed for by the team. The users employ all functions described in section 6.3.2 and mostly adhere to the conventions to which they agreed during the workshops of the project. There are several pieces of empirical data that substantiate this statement.

First, there are the subjective statements of the team members that they make use of ELISE. These were taken in the course of two meetings, one on May 25th 2005 and the other on August 22nd 2005 (cf. appendix 11.1.3 for a description of the meetings). Second, there is a log file of the usage covering a period of about 6 weeks in June and July 2005 during which the team members read tables of content and placed orders (cf. appendix 11.6, Table 40). Third, there are the e-mails in the e-mail archive that were sent out by the student worker in the library reminding the researchers to check for new tables of content. Fourth, there are the e-mails in the e-mail archive that were sent out by the student worker informing the researchers that articles they have ordered are now available. Fifth, there are the e-mails in the e-mail archive that were sent out by me informing everybody that new tables of contents are available in ELISE.

Socio-technical Setting for ELISE

With regard to case study 2 it is possible to resume that the socio-technical perspective that was taken from the very beginning of the project lead to a socio-technical setting in which the usage of the CSCW-system ELISE is embedded into the work procedures of the team. In the following the terms defined in section 3.2.1 are revisited to see how the early theoretical considerations apply to the current situation in case study 2.

A socio-technical setting as defined in Term 22 comprises a CSCW-system and a socio-technical organization. Case study 2 yielded the CSCW-application ELISE and a socio-technical organization to which the research team and the library staff belong as members.

The definition of a socio-technical organization in Term 20 includes two characteristics: first the communicative use of a CSCW-application and second the maintenance of socio-technical self-descriptions. Both characteristics are true for the socio-technical setting created by ELISE and the research team. The communicative use of ELISE includes informing each other about reading and ordering articles, asking the library staff to get articles and recommending articles to other colleagues.

Example of socio-technical self-description	**Form** **(inscribed, documented, ephemeral)**
Group members may see each other's activities in ELISE such as ordering articles or reading tables of content.	This self-description is inscribed in the CSCW-application ELISE. The users' actions such as reading and ordering are recorded in a field which is displayed in the GUI (cf. Figure 25).
The library staff will remind the scientific staff of checking tables of content in ELISE by sending e-mails or addressing them personally.	This self-description is documented in SeeMe diagrams (cf. Figure 37).
It is possible to ask the library staff to wait with processing a set of tables of contents, if one or more researchers did not have the time to read them due to holidays or heavy work load.	This self-description is only ephemeral. It resulted from an e-mail and a subsequent discussion in the group. A researcher sent a mail asking the library staff to wait a few days before processing the orders for articles. Others reacted to the e-mail and in direct conversation, the members of the group agreed to postpone the deadline they had defined before.

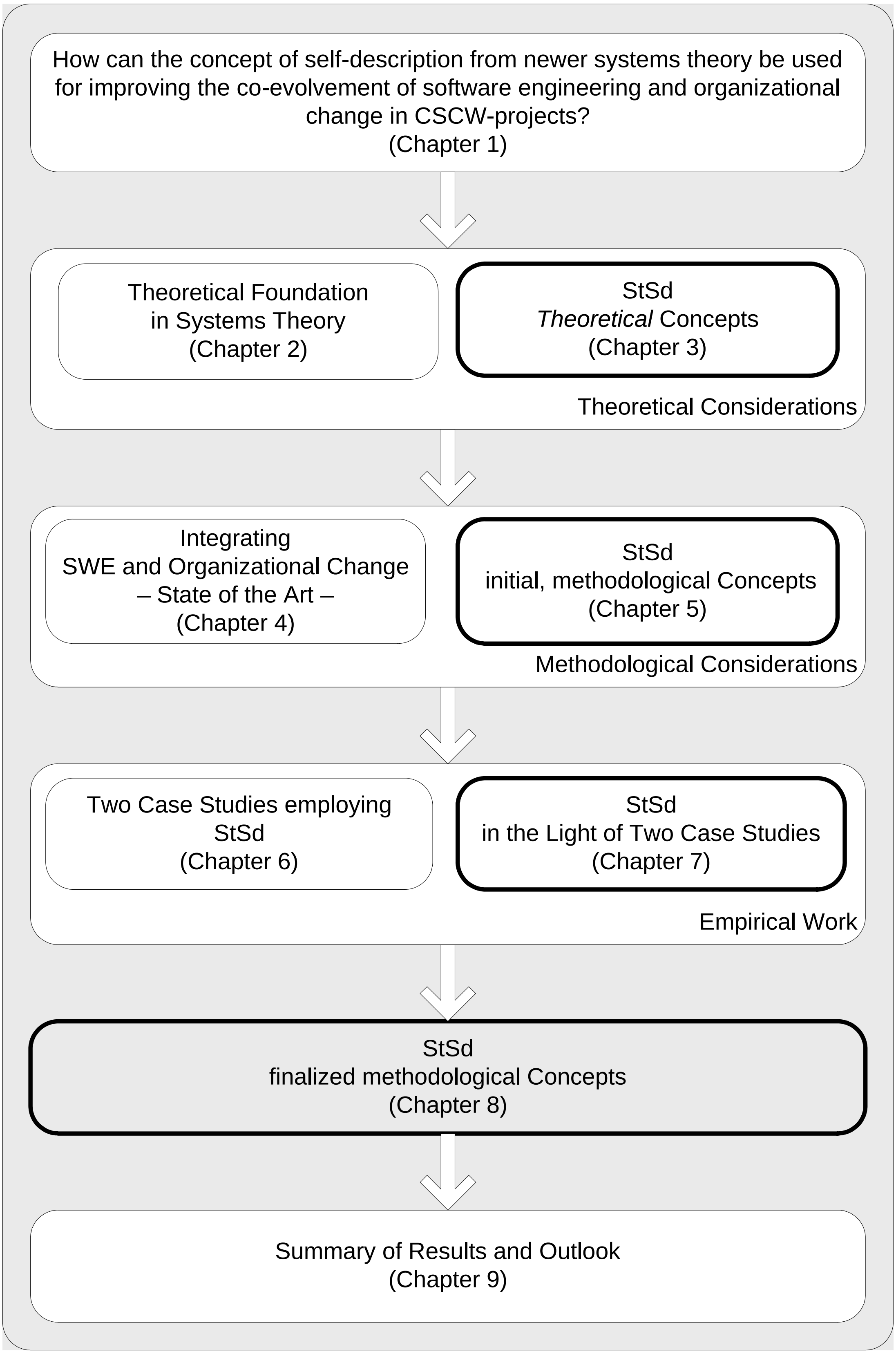
Table 29: Examples of StSd in Case Study 2

The question whether the research group maintains socio-technical self-descriptions requires looking at the definition in Term 21 which gives four characteristics of socio-technical self-descriptions. First, StSd describe which actions are acceptable within the organization and how they relate to the CSCW-system. The research team has indeed developed many such descriptions which have been mentioned throughout the examples discussed in this chapter. Three of them are repeated in Table 29 (left column). The second characteristic in Term 21 says that socio-technical self-descriptions provide guidance for the members of the socio-technical organization. The feedback given by team members which is summarized in Table 22 indicates that the documented forms of StSd are indeed regarded as helpful guidance. Furthermore, the fact that the team members largely adhere to the agreements, can be taken as an indication that the StSd guided the actions of the team members related to ELISE and the task of viewing scientific periodicals.

The third characteristic distinguishes three forms of socio-technical self-descriptions: inscribed, documented and ephemeral which are interrelated as illustrated in Figure 16. Indeed all three forms can be found in case study 2: the examples of self-descriptions in Table 29 are chosen such that there is one example for each form.

The fourth characteristic of socio-technical self-descriptions states that they are communications about communications; socio-technical self-descriptions take place on a meta-level, describing other communications and should be distinguished from other communications. In case study 2 it is possible to identify self-descriptive communications: communications about the usage of the CSCW-application ELISE took place in dedicated workshops, as part of regular team meetings but also in form of e-mails. The workshops have already been discussed extensively, so consider the example of the ephemeral self-description in Table 29: the researcher who asked for postponing the deadline initiated a meta-level communication on the usage of ELISE. The team members briefly negotiated whether everybody agreed to providing a few more days for reading the tables of contents. The resulting self-description (that it is indeed possible to postpone deadlines) is not documented anywhere, but the discussion indicates that ELISE is indeed included into the self-descriptive communications of the group.

To conclude this section, a few words shall be said about the outcome of case study 1. Since the system SpiW-Com was developed as a scientific prototype, it was not used operationally by the "Westkreis"; and therefore the outcome cannot be analyzed as it is done above for case study 2. However, it is possible to compare the status of the two case studies at the time of the qualification workshops (cf. Example 34). At this time the interventions in case study 1 had lead to the same results as in case study 2: there is the CSCW-system (SpiW-Com) and there are documented socio-technical self-descriptions in form of semi-structured diagrams. The participants' feedback to elaborating socio-technical self-descriptions along with the development of the CSCW-system was positive (cf. Example 21). So there is no reason why the "Westkreis" and SpiW-Com should not have evolved to a socio-technical setting if the project had been prolonged.


How can the concept of self-description from newer systems theory be used for improving the co-evolvement of software engineering and organizational change in CSCW-projects? (Chapter 1)
Theoretical Foundation in Systems Theory (Chapter 2)
StSd *Theoretical* Concepts (Chapter 3)
Theoretical Considerations
Integrating SWE and Organizational Change – State of the Art – (Chapter 4)
StSd initial, methodological Concepts (Chapter 5)
Methodological Considerations
Two Case Studies employing StSd (Chapter 6)
StSd in the Light of Two Case Studies (Chapter 7)
Empirical Work
StSd finalized methodological Concepts (Chapter 8)
Summary of Results and Outlook (Chapter 9)

8 StSd – Finalized methodological Concepts

8.1 Basis for the finalized methodological Concept

8.1.1 StSd – what did the initial Concept achieve?

The research presented in this thesis was motivated by the observation that CSCW-projects often fail in the sense that newly developed software is not used at all, used for other purposes, or even sabotaged (cf. section 1.1). In such cases the investment in terms of resources like time, money, or equipment was spent without achieving the desired effect.

This thesis presents two case studies of CSCW-projects in which these problems were successfully addressed. In case study 2 (electronic ToC service) we succeeded in establishing a socio-technical setting as it is described in section 3.2.1:

- The software system ELISE (a database application) was designed and implemented;
- New cooperative work procedures for selecting and ordering articles from scientific journals were planned and established;
- The software is operational and the corresponding work procedures are in effect since May 2005.

Case study 1 (mobile communication system) also yielded a software system together with corresponding work procedures; the system did not go operational because the research project ended, otherwise there was no indication why the establishment of a socio-technical setting should not have been successful here as well.

The successful outcome of the case studies was achieved by keeping a strict socio-technical perspective throughout all phases of the projects. The process of planning and establishing new work procedures that make use of the CSCW-system was interwoven with the process of software engineering. Beside this general stance on CSCW-projects, graspable methodological concepts and techniques were employed to reach the goal. The organizations were supported in explicitly planning and documenting the way they intended to integrate the CSCW-system into their work procedures. The resulting documentation is one form of the organization's socio-technical self-description (StSd).

How can the insights gained in the case studies be made available for the field of CSCW? Section 8.2 provides a set of recommendations derived from the combination of theoretical considerations and empirical experience. These recommendations help to keep a socio-technical perspective throughout a CSCW-project; they do not form a complete and closed method for conducting CSCW-projects. The recommendations are meant to be applied in combination with other techniques and methods known in the field of CSCW or software engineering.

Sections 8.1.2 and 8.1.3 provide the basis for the recommendations by summarizing the main results from the stages in which I elaborated the concept of socio-technical self-descriptions in this thesis.

8.1.2 StSd – Theoretical Elaboration of the Concept

Chapter 2 started out by introducing Luhmann's system theory and I derived its explanatory potential for phenomena known in CSCW-projects as well as its potential for designing methodological support for CSCW-projects. With regard to the methodological support of CSCW-projects, the most important terms discussed in chapter 2 are structural coupling (cf. section 2.3.7, Term 10) and self-description (cf. 2.4, Term 11). I use Luhmann's concept of structural coupling for describing the relationship between an organization and a CSCW system. For Luhmann's concept of self-description I derive 4 characteristics which can be empirically observed and which are the basis for my concept of socio-technical self-descriptions.

Chapter 3 presents my comprehensive concept of socio-technical self-descriptions (StSd). It is explained how StSd fit into the general context of an organization making use of a CSCW-system; its relation to other concepts present in the field of CSCW are discussed. With regard to the methodological support of CSCW-projects, the most important results of chapter 3 are:

- The definition of a socio-technical setting (cf. Term 22) as the outcome of a CSCW-project;
- The precision of the terms socio-technical self-descriptions and socio-technical organization (cf. Term 20 and Term 21);
- A process model for CSCW-projects which is the basis for keeping a socio-technical perspective throughout all phases (cf. section 3.2.2, Figure 11).

Based on the theoretical insights and concepts of chapters 2 and 3 in combination with best practices identified from the analysis of available methods (chapter 4), chapter 5 introduces my initial concepts for the methodological support of CSCW-projects. At the center of the methodological support there are the socio-technical self-descriptions: the members of the organization are encouraged and supported to engage in self-descriptive communications regarding the new CSCW-system. Thereby they should be enabled to plan computer supported work procedures before the CSCW-system is deployed.

My initial methodological concepts were employed in two exploratory case studies (cf. chapters 6 and 7) with the goal of gaining further insights by analyzing the empirical evidence. The analysis was guided by a set of research questions (cf. Table 14) and yielded results regarding the continuous integration of organizational change and software engineering within a CSCW-project. With respect to socio-technical self-description further insights were gained about their content, suitable forms for documentation and their creation and discussion within facilitated workshops.

8.1.3 StSd – Empirical Results from Case Studies

In the following the research questions of Table 14 are revisited for summarizing the insights of the case studies. Chapter 7 contains a section for each of the questions; there each question is discussed in detail, using empirical evidence from the case studies. Here, these discussions are recapitulated, summarizing the main results. The recapitulation starts at the end – with the last research question concerning the outcome of the CSCW-projects in the case studies.

I2: Were the CSCW-projects successful in creating socio-technical settings?

Yes, in case study 2 (electronic ToC service) a socio-technical setting evolved that includes the CSCW-system ELISE and a socio-technical organization comprising the scientific team of 8 researchers and a student worker from the library. The technical development of ELISE was intertwined with planning new work procedures for the scientists as well as for the library staff; socio-technical self-descriptions including the work procedures and additional conventions regarding the use of ELISE were discussed, agreed to and documented in form of semi-structured diagrams. Since May 2005 the new work procedures are in effect and ELISE is operational.

The CSCW-system SpiW-Com in case study 2 (mobile communication system) was not deployed and used, because the prototype was a research prototype and the project ended before the prototype could be deployed. But as the other case study, case study 1 yielded the CSCW-system SpiW-Com together with socio-technical self-descriptions in form of semi-structured diagrams; these document how new work procedures would make use of SpiW-Com and contain additional conventions necessary for ensuring the coordinated use of the CSCW-system.

I1: Can artifacts of socio-technical self-description relate the two processes of software engineering and organizational change by serving as boundary objects?

Yes, in both case studies, the socio-technical self-descriptions that were documented in form of semi-structured diagrams were used as boundary objects between different groups of stakeholders in the project. Two pairs of groups are of special interest:

- StSd served as boundary objects between users and designers;
- StSd served as boundary objects between heterogeneous groups of users.

StSd can be an artifact shared by users and designers because they comprise information needed for the technical design as well as for organizing new computer supported work procedures. But equally important, they include information related to the work procedures of different groups of users. In case study 1, drivers and dispatchers documented their respective needs in the diagrams; in case study 2 this was true for the scientific team on the one side and the student worker in the library on the other.

Furthermore the StSd in form of semi-structured diagrams served as a guiding artifact through all phases of the CSCW-projects. Their initial versions were conceived in the analysis of the status quo from which the first ideas for new computer supported work procedures were derived; from then on the diagrams documented the advancing projects by reflecting organizational decisions as well as the current state of the software development.

Some limitations regarding the function of StSd as boundary objects were identified; these are reached when constraints regarding the form or content of StSd are induced by a group; and when these constraints compromise their suitability for capturing the organization's socio-technical self-description.

C1: What are the contents of documents that contribute to the self-description of the socio-technical organization with regard to the new CSCW-system?

Socio-technical self-descriptions document computer supported work procedures by describing the work procedures, the CSCW-system, their relation and additional agreements necessary for coordinating the usage of the CSCW-system. An analysis of all diagrams created in the course of the two case studies yielded the following thematic clusters for StSd:

- Work procedures
- Relations between work procedures and software-functions
- Agreements concerning the original cooperative task
- Agreements concerning the usage of the CSCW-system
- Usage of additional technical systems
- Meta-level comments concerning the project

C2: Do the topics identified in C1 differ from the scope of other documentations such use-cases, business process models or scenarios?

Yes, StSd are unique concerning the composition of their content. None of the other types of documentations that are common in software engineering or organizational change projects aim at describing a socio-technical setting. Use cases specify the software system by completely specifying the range of possible interactions; the list of topics to be included into StSd exceeds this purpose. Business process models describe business processes on a level that is more abstract than the level needed for work procedures; an observation from the case studies is that the abstract level of business processes is often inscribed into the CSCW-system itself. StSd provide additional information for creating viable work procedures which make use of the CSCW-system

C3: How did the topics that were eventually documented emerge during the project?

Since the case studies were conducted as action research projects, answering this question was important with regard to the validity of the results. If all topics documented in the StSd were brought up by me in my role as a facilitator or participant of the workshops, I could not argue that the content of StSd was determined by process of structural coupling between organization and CSCW-system. But the analysis of the case studies provides clear evidence, that the participants of the workshops contributed to the content of StSd themselves.

C4: What importance did the participants attach to the contents of socio-technical self-descriptions?

Like question C3 this research question was also included to compensate my dual role as researcher and participant in the case studies. I needed to make sure that the topics included in the StSd were not just documented because I told everybody to do so. In questionnaires and interviews the participants of the case studies were asked how they evaluated the content of the StSd. The results were unanimously positive; all stakeholders involved – users, managers and software engineers – agreed that the StSd contain information helpful and important for the successful usage of the CSCW-systems.

The subjective statements from the participants of my case studies are backed by additional evidence from a controlled experimental field study. It was possible to show that supporting the discussion and documentation of StSd not only increases the number of agreements users make with regard to the usage of a knowledge management system, it also intensifies the usage of the system.

F1: Were semi-structured diagrams suitable for documenting the topics discussed during the project?

Yes, the analysis of both case studies showed that semi-structured diagrams were able of capturing all topics discussed regarding the usage of the CSCW-system. By exploiting the options of the modeling notation SeeMe for representing uncertainty, critical aspects of formal representations of work procedures could be avoided. In particular, the contingency of human action with regard to the primary work task as well as with regard to the usage of a CSCW-system could be documented. An important lesson learnt from the case studies is, that one needs to take care not to hide important issues behind notational details. The most efficient way of modeling a certain issue is not necessarily the way that represents the weight of the issue best.

The enrichment of the diagrams by integrating further material proved to be advantageous. The additional material is connected to elements in the semi-structured diagrams by hyperlinks; in the case studies this included:

- Photos of offices, shelves, equipment etc. taken during the fieldwork are linked for creating a more vivid description of the work context;
- Photos of documents or forms are linked for providing concrete examples of information exchanged during the work processes;
- Specification documents created in the course of the project are linked for providing further information;
- Screenshots of the GUI are linked for setting interactive functions in relation to steps in the work procedures;
- If part of the CSCW-system is realized as a web interface, this is linked to the work procedures such that in can be invoked directly.

The participants of the case studies were asked to evaluate how suitable semi-structured diagrams are for documenting the topics discussed; the feedback was consistently positive.

F2: How is the software-system represented in the socio-technical diagrams and how does this representation change in the course of the project?

Within a StSd the representation of the software-system needs to complement the descriptions of the work procedures such that the usage of the software can be discussed and planned; therefore a CSCW-system within a StSd is described from a user's point of view. However this description does not remain the same during all phases of the project; it is adapted as software engineering proceeds. In the case studies I identified three stages: first, the diagrams reference artifacts such as paper forms that will be replaced by the CSCW-system; second, the diagrams describe the CSCW-system using symbols of the modeling notation augmented by links to specification documents; third, the diagrammatic or textual description of the CSCW-system is replaced by links to screenshots or even the CSCW-system itself.

This adaption serves two purposes: First, it supports the future users in planning their work procedures by providing them with the most recent results from the technical development. Second, it allows the validation of each result and iteration provided for the CSCW-system in the organizational context of work procedures.

The case studies showed that once the CSCW-system is used, it is helpful to include links from the CSCW-system to the diagrams; this way they can serve as an additional helping resource within the CSCW-system. So in a fourth phase the unidirectional way of including references from the diagrams to the CSCW-system is extended to bidirectional references.

WS1: What is the experience with electronic, semi-structured diagrams as guiding artifacts for the facilitation of workshops?

The case studies showed that semi-structured diagrams representing work procedures and how they make use of a CSCW-system are helpful in structuring participatory workshops in the context of CSCW-projects. The diagrams supported a socio-technical perspective by structuring the discussion in all phases of the projects according to the context of use rather than the features of the technical system; but within this structure technical as well as organizational issues were discussed.

However, the case studies also showed that the rather strict walkthrough-concept with predefined questions had to be relativized and replaced by the notion that the diagrams serve as guidance throughout the workshops rather than as a script. First, it was not possible to identify a fixed set of walkthrough questions, because the need of the different projects and in fact of every workshop were too heterogeneous. The experience does indicate that the question guiding the workshop should be a socio-technical one in the sense that it addresses organizational as well as technical issues; and suggestions can be derived for certain types of questions that proved to be helpful. Second, the creative and constructive aspects inherent to a CSCW-project require more flexibility than walkthroughs that are applied as inspection methods. When a STWT workshop is held to invent, discuss and agree upon the usage of a CSCW-system, then the discussion should not be restricted by forcing it along the work procedures; but when it comes to documenting the results, then it is helpful again to step through the diagram systematically, to make sure that no part of the work procedures was forgotten.

Both case studies also showed that documenting socio-technical self-description in form of semi-structured diagrams causes significant cognitive effort on everybody – facilitator as well as participants. However, the positive feedback to the documentation that was available at the end of the case studies indicates that this cognitive effort should be invested. In the case studies several options – technical as well as organizational – for alleviating the additional cognitive effort were tried. On the organizational side, it was helpful

- to ensure that participants as well as facilitator have sufficient knowledge of the modeling notation;
- to include a draftsman into the facilitation team who takes over the task of editing the diagrams during discussion;
- to design the workshop such that breakout groups are formed and that they create diagrams as part of their presentation to the whole group.

The technical support in the case studies consisted of the SeeMe editor which is a dedicated editor supporting the editing of semi-structured diagram and providing special functions for presenting complex diagrams. Using the SeeMe editor enabled me to identify further design recommendations for technical support regarding the following four settings:

- Collaborative editing during workshops (Co-located Collaboration)
- Collaborative reading during workshops (Presentation)
- Individual editing between workshops
- Individual reading between workshops

WS2: What is the experience concerning the integration of organizational and technical issues within the workshops?

All STWT-workshops conducted during the case studies integrated the discussion of technical issues, organizational issues and their interrelations. The experience was positive and yielded good results. But the case studies did show that balancing technical and organizational perspectives within one workshop requires vigilance and the facilitator's intervention. Especially in case study 1 there was a tendency to close a discussion once the technical issues are settled, and to assume that open organizational decisions would be taken "elsewhere". The lesson to be learned for software engineers participating in STWT-workshops is that they have a dual role: on the one side they are software engineers constructing the CSCW-system and needing to learn about the requirements. But on the other side they are software engineers providing technical expertise for the future users elaborating their future work procedures. And the latter role requires them to participate in discussions even if all questions they need for their technical design have been answered.

The lesson to be learned for the facilitator is that she needs to pay attention to situations in which organizational issues are put aside and to intervene in order to give them the room they need.

8.2 Recommendations for Supporting StSd in CSCW-Projects

The following recommendations provide the essence of all considerations regarding the concept of socio-technical self-descriptions in the context of CSCW-projects. Their application in CSCW-projects should support the project team to keep a socio-technical perspective throughout the whole project and to support the evolvement of a socio-technical setting in which new computer supported cooperative work procedures make sensible use of a new (or changed) CSCW-system.

Figure 59 is an extension of Figure 11 and shows a project model for CSCW-projects including STWT workshops and socio-technical self-descriptions. The recommendations explain and refer to the model.

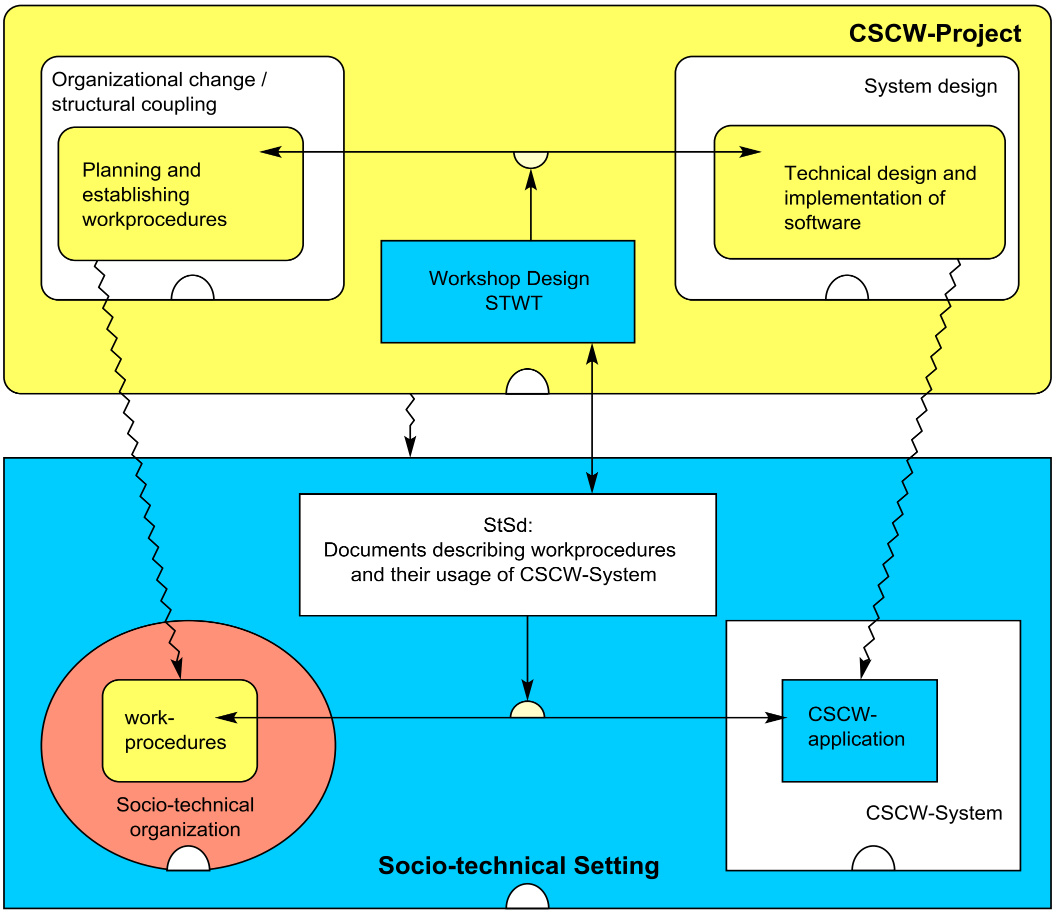

CSCW-Project
Organizational change / structural coupling
Planning and establishing workprocedures
System design
Technical design and implementation of software
Workshop Design STWT
StSd: Documents describing workprocedures and their usage of CSCW-System
work-procedures
Socio-technical organization
CSCW-application
CSCW-System
Socio-technical Setting

Figure 59: Locating StSd and STWT in a Process Model for CSCW-Projects

Recommendation 1: Regard the result of a CSCW-project as a socio-technical setting which is bipartite. On the one side there is the technical product, the CSCW-application, and on the other there are new work procedures that incorporate it.

Although social and technical aspects are highly interwoven, it is helpful to keep in mind at all times that a feature in the CSCW-application must be matched by a description how to include it into the cooperative work procedures; and that both parts require deliberate action for their evolvement.

The process model in Figure 59 represents the *socio-technical setting* in the lower part. It includes the two aspects and their interrelation: *work procedures* are represented as an activity embedded into the context of a *socio-technical organization*. *CSCW-application* is a technical entity embedded into a larger technical *CSCW-system*.

Related sections:

- Terms: CSCW-project (Term 23, p.85); CSCW-application (Term 8, p. 35); CSCW-system (Term 7, p. 35); socio-technical organization (Term 20, p. 82); socio-technical setting (Term 22, p. 83).
- Figure 6 and section 3.1: The process model in Figure 59 contains a reduced representation of a socio-technical setting. Figure 6 is a more detailed representation and section 3.1 discusses the theoretical concepts behind it.

Recommendation 2: Regard the CSCW-project itself as bipartite. On the one side there is the organizational process leading to new cooperative work procedures; on the other there is the technical process leading to a new (changed) CSCW-application.

The bipartite result needs a correspondence within the CSCW-project, because the establishment of new work procedures requires different methodological support than the design and implementation of a CSCW-application. While engineering methods that systematically lead to technical design and implementation are adequate for software, organizations cannot be changed using the same methods. It is a helpful concept to view organizations as autopoietic social systems which can only change from inside. Therefore the process of *planning and establishing new work procedures* includes communication among the future users: discussions for developing a common understanding of the CSCW-system; negotiations about different options for its usage; conventions for its usage; commitments to use the CSCW-system according to the conventions. The support of these communications is not covered by the process of system design that produces a new CSCW-system.

Therefore, the *CSCW-Project* represented in Figure 59 comprises two subprocesses: *Organizational change / structural coupling* and *System design*. Since both are very broad processes, a sub process is identified for each one of them to indicate precisely what the methodological support for socio-technical self-descriptions can achieve. Within *Organizational change / structural coupling* the relevant sub process is *Planning and establishing work procedures*. Within *System design* the focus lies on the *technical design and implementation of software*.

The process model of Figure 59 should not be misinterpreted as suggesting that a CSCW-project can be split up into two parts. The two subprocesses are highly interrelated and only together they form the overall CSCW-project. In practice meetings, workshops or documents often pertain to both processes. But because they are so highly interwoven it is even more important to emphasize that both parts exist and need to be covered within a CSCW-project. The following recommendations will provide guidelines for doing so.

Related sections:

- Terms: Software engineering (Term 6, p. 34); structural coupling (Term 10, p. 46); CSCW-project (Term 23, p. 85); organizational change (Term 12, p. 58); convention (Term 14, p. 68); Use of a CSCW-application (Term 16, p. 76).
- Concept of organizations as autopoietic social systems: section 2.3.2 introduces Luhmann's conceptualization of social systems as being autopoietic; section 2.3.6 describes organizations as one special type of social systems.
- Concept of structural coupling: section 2.3.7 explains Luhmann's original concept; section 3.2.1 discusses the transfer into the context of CSCW-systems and compares it to the concepts of adoption and appropriation (p. 83).
- Theoretical reasoning for distinguishing between structural coupling and technical design and implementation of software: sections 2.2 and 2.3.
- Systemic intervention: section 4.2 introduces systemic intervention as a discipline that uses insights from Luhmann's theory for providing methods supporting organizational change.

Recommendation 3: Support the organization in its process of structural coupling by creating documents of socio-technical self-description.

Luhmann devised the concept of self-description to explain how organizations can create and keep an identity. In contrast to other types of systems such as biological systems, organizations do not possess anything physical – like a membrane – that constitutes their boundary to the environment. Instead, the members of an organization must engage in a continuous process of negotiation deciding which actions they expect from each other and which actions are not acceptable within the organization.

Self-descriptions serve two purposes in this context: First, they guarantee stability. Self-descriptions define the boundaries of the organization by describing the expectations towards the behavior of its members. Second, self-descriptions enable change. By explicitly describing the organization, self-descriptions can be subject to reflection and debate leading to change.

Socio-technical self-descriptions (StSd) serve the same purpose, but they explicitly describe how the use of a CSCW-system is integrated into the organization's work procedures. By supporting the creation and maintenance of socio-technical self-descriptions within a CSCW-project, the organization is first supported in changing its work procedures to make use of the CSCW-system; second it is supported in maintaining stability in using the CSCW-system.

There is one distinct advantage of creating explicit socio-technical self-descriptions over uncoordinated processes of adoption: it increases the organization's chance to develop elaborated ways of using the CSCW-system.

Related sections:

- Terms: Self-description (Term 11, p. 57); socio-technical self-description (Term 21, p. 82)

- Section 2.4 introduces Luhmann's original concept of self-description from which I derive four characteristics that can be used for designing supporting methods.
- Section 3.1 places the concept of socio-technical self-descriptions into the larger context of socio-technical settings; here the relations between other concepts known in CSCW and socio-technical self-descriptions are discussed.
- Section 3.2.1 derives the definition of socio-technical self-descriptions (Term 21).

Recommendation 4: Check the 6 categories of contents in socio-technical self-descriptions for their relevance in your project and include them in the StSd accordingly.

Work procedures (cf. section 7.1.1.1, pp. 170)	
	Include all tasks that are of importance to the participants, especially in the beginning when it cannot be foreseen whether and how they might be related to the CSCW-system (cp. Example 1, p. 171; Example 2, p. 172). Be aware that the description of the work-procedures will change as the project proceeds. Activities will loose or gain importance as the CSCW-system develops; and accordingly activities will need to be taken away from or added to the diagrams (Example 2, p. 172).
Relations between workrelated activities and software-functions (cf. section 7.1.1.2, pp. 175)	
	Relate software-functions to activities relevant in the work-procedures rather than introducing extra activities for handling the CSCW-system (Example 4, p. 175; Example 5, p. 176). There should be an increase in granularity from the description of work-procedures to the description of software functions (Example 4, p. 175; Example 5, p. 176; Example 31, p. 218). Socio-technical self-descriptions are no interaction diagrams. Name conditions under which certain software-functions will be used (Example 7, p.179). Include information about software-functions that are available but will not be used (cp. Example 6, p. 178).
Agreements concerning coordination of primary task (cf. section 7.1.1.3, pp. 180)	
	Describe how the participants agree to cooperate with each other; especially when there are issues of responsibility that might be affected by the new CSCW-system (Example 9, p. 181). If the CSCW-system includes a workflow, check whether there are tasks to be done in addition to those directly requested by the CSCW-system (Example 11, p. 183).
Agreements concerning the Usage of the CSCW-system (cf. section 7.1.1.4, p. 184)	
	Do not assume that the usage of a CSCW-system is self-evident and unambiguous. Additional agreements (conventions) are needed and should be documented in the course of the project (Example 12, p. 184; Example 13, p. 184).
Usage of additional technical systems (cf. section 7.1.1.5, pp. 186)	
	There will be other technical systems with overlapping functionality. Describe the usage of these systems in relation to the new CSCW-system (cp. Example 14, p. 186; Example 15, p. 188). There will be other technical systems with complementary functionality. Describe the usage of these systems within the work procedures that are relevant for the new CSCW-system (cp. Example 11, p. 183).
Meta-level comments concerning the project (cf. section 7.1.1.6, pp. 189)	
	Open Issues that require further discussion and decision need to be documented. If such issues are related to the technically supported work procedures, use comments in the diagrams (Example 8, p. 179; Example 16, p. 189).

Table 30: Detailed Recommendations concerning the Contents of StSd

The analysis of the case studies provided 6 categories of content for socio-technical self-descriptions. Table 30 lists the categories and includes further, detailed recommendations for each of them.

Related sections:

- Identification of categories for content of StSd: section 7.1.1, pp. 169.
- Discussion of each category: sections 7.1.1.1 - 7.1.1.6.
- Deriving the content of self-descriptions from Luhmann's theory of social systems: section 2.4.5, pp. 56.
- Transfer from social to socio-technical self-descriptions by including the usage of a CSCW-system as a topic: section 3.1.2, pp. 64

Recommendation 5: Use semi-structured diagrams interweaving additional material for documenting the organization's socio-technical self-description.

In order to make a self-description durable within an organization, it needs to exist in form of documents. Therefore, document socio-technical self-descriptions in form of semi-structured diagrams that

a) combine the representation of work-procedures with that of the supporting CSCW-system;

b) interweave other material describing the working environment and the CSCW-system such as photos, documents or screenshots by defining links.

Figure 60 proposes a template for socio-technical self-descriptions documented as semi-structured diagrams.

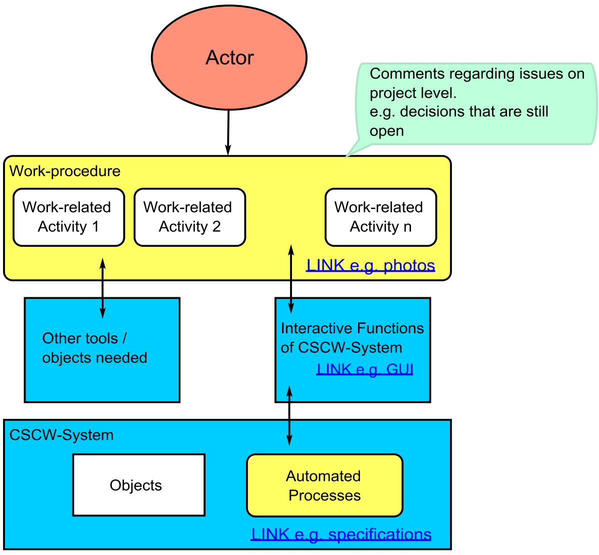


Figure 60: Template for Illustrating CSCW

At the top of the diagrams there are the actors, directly below the work-procedures they conduct. The work-procedures can be illustrated e.g. by photos giving an impression about the surroundings. The representation of the CSCW-system follows the paradigm of separating an interactive layer from an internal layer. The interactive functions are directly related to the work-procedures and can (indeed should) be linked to artifacts like GUI screenshots or prototypes. The internal functions describe what the user can expect from the CSCW-system, static elements that are stored by the system can be identified as well as automated processes. The description of the CSCW-system's internal functions can be detailed by links to further specification documents. Comments are used for project-related issues such as decisions that have been made or still need to be made.

Table 31 contains further, more detailed recommendations concerning the employment of semi-structured diagrams.

Representing the situated character of work procedures	
	Include the contingency of human action in the diagrams by making use of appropriate notational elements - for the work procedures themselves (Figure 15, p. 127; Figure 27, p. 171; Figure 33, p. 179). - for the incorporation of the CSCW-system into the work procedures (Figure 18, p. 140; Figure 30, p. 176; Figure 31, p. 177).
Representing the CSCW-system	
	There are three phases during CSCW-projects that require different representations: - First, during the requirements elicitation there artifacts like paper forms that will (or might be) replaced by the CSCW-system. Integrate them into the semi-structured diagrams in form of pictures whenever possible. Including pictures will give a better visualization of how the artifacts are used than translating them into notational elements (cf. Example 29, Figure 50). - Second, when new ideas about the CSCW-system are discussed these are documented in form of notational elements within the diagrams (cf. Example 29, Figure 51). - Third, when screenshots or prototypes and eventually the final CSCW-system itself are available, they are linked into the diagrams (cf. Example 29, Figure 53) Avoid redundancy when describing the CSCW-system. Aspects that are described in form of notational elements in the diagrams in early phases of a CSCW-project need to be replaced by links once they have been implemented (Example 29, p. 214; Example 31, p. 218).
Further advice	
	- Select those elements from the modeling notation that emphasize important issues, rather than those that might optimize the size of a diagram but hide essential information behind notational details (Example 25, p. 208). - Be aware that problems may arise if the organization already uses modeling languages e.g. for their business processes. The recommendation is to keep the diagrams for socio-technical self-description apart from other diagrams that serve other purposes in the organization if these other purposes interfere with the purposes of socio-technical self-descriptions. Use links to provide references to other organizational documentation where appropriate (Example 26, p. 210). - Once the CSCW-system has been implemented, include links from the GUI of the CSCW-system to the semi-structured diagrams. This way the diagrams can server as an additional helping resource for using the CSCW-system (Example 32).

Table 31: Detailed Recommendations for employing semi-structured diagrams

Related Sections:

- Section 4.3.5 discusses the chances and risks of using formal methods for describing work procedures. Semi-structured diagrams are introduced as one way of keeping the benefits and avoiding the downsides.
- Documents are only one form in which socio-technical self-descriptions occur. Section 2.4.2 (pp. 50), section 3.1.3 (pp. 69) and section 5.1.2 (pp. 134) discuss the three forms (ephemeral, documented, inscribed) and their interplay.
- The overview of diagrams in appendix 11.3 (pp. 307) provides an abundant source of examples.
- Section 11.2 in the appendix provides an overview of the SeeMe notation which was used in both case studies.

Recommendation 6: Use a series of STWT-workshops to provide a communicative context for the creation and maintenance of socio-technical self-descriptions.

Socio-technical self-description is a social phenomenon, it takes place only if members of the organization communicate with each other and come to agreements which are then documented. Therefore, support the organization in self-descriptive communications regarding the usage of the CSCW-system. STWT-workshops have proven to be good setting for this purpose. Table 32 provides detailed recommendations with regard to STWT-workshops.

Facilitation	
	- Basic facilitation skills are the prerequisite for conducting STWT (cf. section 5.2.3) - Use semi-structured diagrams representing the organization’s socio-technical self-description for structuring the discussion; but be aware that creative discussions may not move along the sequence in the diagrams (cf. Example 37, p. 225). - Prepare guiding questions to “walk through” the diagrams; take care that these questions are “socio-technical” in the sense that they address the relation between social and technical aspects (cf. Example 36, p. 224). - Use semi-structured diagrams for documenting results (cf. Example 16, p. 189). - Be aware that although the semi-structured diagrams should be a guiding artifact throughout the project, there are situations in which they are not the appropriate means of documentation. Discussing characteristics of the CSCW-system that do not directly relate to work procedures is one example, discussing and deciding among a large number of options another. Be flexible to switch to other methods of documentation or visualization during workshops (Example 24, p. 206). - Combine the discussion of vivid scenarios with documenting abstract work procedures that can serve as socio-technical self-descriptions (Example 37, p. 225).
Integrating the processes of structural coupling and system design within one workshop	
	- Be explicit that the goal of the project is the socio-technical setting comprising the CSCW-system and the corresponding work procedures in the organization; provide room for organizational as well as technical discussions (Example 44, p. 240). - Integrate the organizational perspective into phases of the project that tend to concentrate on the CSCW-application itself or on individuals using it (Example 33, p. 222; Example 34, p. 223). - Be explicit that the task of structural coupling – i.e. the integration of the CSCW-system into work procedures – must be performed by the organization itself. Take care that software engineers are present for providing technical expertise but make sure that organizational consequences are agreed to among the members of the organization (Example 43, p. 239).
Electronic tool support	
	- Use an editor for creating the diagrams, because it makes the integration of additional material, the continuous modifications and the distribution easier (cf. Example 38, p. 226). - Use an editor that not only provides general drawing functions, but one that supports the following four settings: collocated collaboration, presentation, individual editing, and individual reception. - Be sure that not only the facilitator and the draftsman but also some of the other participants have expertise in handling the editor. This makes you more flexible in delegating the task of editing diagrams during and in-between workshops (cf. Example 42, p. 232). - Choose ways of editing diagrams such that the participants accept them as “their own”. If the facilitator or a draftsman edits the diagrams, you must ensure that the editing is tightly connected to the discussion (Example 39, p. 228; Example 42. p. 232)
Alleviating the cognitive effort of creating and maintaining diagrams	
	- Be explicit that creating and maintaining socio-technical self-descriptions in form of semi-structured diagrams does not “happen by itself”, but causes cognitive effort during the workshops (cf. Example 40, p. 228; Example 41, p. 229). - Motivate the additional effort by explaining the benefits a documented socio-technical self-description brings (cf. Example 20, p. 199; Example 27, p. 212; Example 28, p. 212). - Provide enough training in the modeling language to reduce the cognitive effort of understanding them. - Include a draftsman into the team, if the task of facilitation can be expected to be difficult (cf. Example 39, p. 228). - Delegate modeling into subgroups during a workshop (Example 42, p. 232).

Table 32: Detailed Recommendations for Conducting STWT-Workshops

Related Sections:

- Section 3.1 describes the complex socio-technical setting that develops when an organization uses a CSCW-system.
- Terms: CSCW-project (Term 23); socio-technical setting (Term 22).
- Section 4.2 introduces systemic intervention and its concept of supporting but not determining organizational change.
- Section 5.2.3 introduces the workshop design STWT, refers to known "walkthroughs" and also summarizes basic information about facilitation.
- Section 7.3 discusses the experience with STWT-workshops in the case studies.
- Section 4.3.3 (pp. 120) introduces the concept of scenario based design. Although socio-technical self-descriptions aim to provide a more abstract description than scenarios, ideas from scenario based design can be interwoven into the workshop design.
- Section 7.3.1 (pp. 234) discusses the design implications I derive for editors of diagrammatic models.

Recommendation 7: Use socio-technical self-descriptions as boundary objects helping to maintain a socio-technical perspective throughout the whole CSCW-project.

The goal of a CSCW-project is establishing a socio-technical setting; socio-technical self-descriptions can help to keep the project's focus on this goal.

Use StSd as continuously advancing, guiding artifacts through all phases of the project – including deployment and use (cf. Example 29, p. 214; Example 35, p. 223). This helps to ensure that neither technical nor organizational issues are discussed and decided upon without reflecting their interdependence.

Use semi-structured diagrams documenting StSd as boundary objects between heterogeneous groups involved in the project. Two pairs of groups are of special importance:

- support communication between users and designers;
- support communication between different groups of users.

Be aware that the function of the diagrams as socio-technical self-descriptions and boundary objects is not self-evident (cf. Example 45, p. 244); be sure to make this purpose explicit.

Accept the limits of using StSd as boundary objects (cf. Example 46, p. 245). If there are purposes (e.g. modeling a workflow for steering a workflow engine; modeling an abstract business process applying to a larger number of organizations) that require changes in the diagrams which would compromise their suitability as socio-technical self-descriptions, separate documentation should be created for those other purposes.

Related Sections:

- Section 7.4.1 presents and discusses the experience with StSd as boundary objects in the case studies.

- Section 7.1.2 discusses the differences between StSd and other documents (use case descriptions, scenarios, business processes).

8.3 Integration of StSd and other Methods

The framework for socio-technical self-descriptions I elaborated in this thesis is self-contained in so far as it includes a theoretical part deriving and describing the concept of socio-technical self-description accompanied by a practical part providing methodological support for its employment in the context of CSCW-projects. It is not self-contained in so far as it does not address all aspects of conducting a software engineering project in the context of CSCW. Therefore it is important that the recommendations given in section 8.2 can be incorporated into existing methods guiding the overall organization of CSCW-projects.

I elaborated the recommendations in section 8.2 in a way that they are compatible with other process models. The following sections describe how they could be integrated into the RUP and MUST. I selected RUP and MUST because they are prominent representatives for the field of software engineering on the one hand (cf. section 4.1) and the field of CSCW on the other (cf. section 4.3.3).

8.3.1 StSd and the RUP

The RUP shall be taken as an example to demonstrate how socio-technical self-descriptions can be incorporated into a process model for software engineering. Table 11 in section 4.4 identifies characteristics of the RUP which allow relating the processes of software engineering and organizational change. Among these are: the iterative process model which produces intermediate results to be discussed and which allows the project team to react to changes; the usage of visual models suitable for supporting communicative processes; the idea of having a guiding artifact throughout all phases that include the users' perspective; the inclusion of business models for describing the context of use; the inclusion of a deployment discipline.

In the following I sketch the modifications that would have to be applied to the RUP as it is defined in [Kruchten (2004)], if socio-technical self-descriptions according to the recommendations in section 8.2 were to be used.

There appear to be three options for including StSd into the RUP. First, one could explicitly anchor the socio-technical perspective in all disciplines (cf. section 4.1.3 for a brief introduction of the RUP, its disciplines and phases). In this case the STWT-workshops would be conducted in the requirements discipline as well as in the business modeling or deployment discipline. The StSd as diagrammatic artifacts would be shared among the disciplines. The advantage of such an approach would be that the idea of a socio-technical perspective was deeply anchored across the disciplines. The disadvantage however would be that in the logic of the RUP there would be nobody responsible for keeping the socio-technical perspective throughout the whole project and for maintaining the diagrams as guiding artifacts. Within the RUP artifacts, tasks and roles are organized in form of disciplines. Therefore the second idea is to add a 10th discipline, named the "structural coupling discipline" that takes the responsibility of supporting the creation and maintenance of socio-technical self-descriptions. The disadvantages of the first option would be cured but the new discipline would appear like an odd add-on with unclear interfaces to the other disciplines. So, the third option is to look for a discipline already existing in the RUP and add the responsibility for socio-technical self-description. When looking at the workflows, tasks and artifacts described for the nine disciplines, no single discipline really offers itself for including socio-technical self-descriptions: the business model discipline creates models which could be related to StSd, but its focus is unclear (cf. section 4.1.6) and it is activity is reduced to almost zero in the transition phase. The requirements discipline is responsible for creating the guiding artifacts of the RUP, the use cases and the use case model. However, its purpose is to produce a specification of the software not of the context of use, and it is also has almost no part in the transition phase. The deployment discipline explicitly addresses the usage of the software and its major activity is located in the transition phase. However, it is not active in the early phases of the project, and the tasks and artifacts described for this discipline in the RUP concentrate on the technical product rather than on its context of use.

From these options which all have their advantages and disadvantages, I would choose to integrate the support for StSd into the deployment phase of the RUP. The goals that Kruchten describes for the deployment discipline come closest to the intentions of my framework for StSd. Reconsider the following quote [Kruchten (2004), p. 237]: The deployment discipline comprises

> *„the type of activities that take place when the software product is deployed in its operational environment and of the additional artifacts that may need to be developed for its successful use."*

The following modifications would be necessary to include the support for StSd into the deployment phase:

- Change in timing: the deployment phase needs to start in the inception phase, together with the disciplines business modeling and requirements. It may be reduced during the construction phase, when the focus is on technical issues. Then it needs to be active again in the transition phase just as it is foreseen in the RUP (cf. Figure 12); then however it may not stop but needs to continue until the end of the project and into the phase of use (which is not included in the RUP).
- Additional tasks: the deployment phase needs to include activities for supporting the creation and maintenance of StSd. The main task is to conduct a series of STWT-workshops according to the recommendations given in section 8.2.
- Additional artifacts: the semi-structured diagrams which document the StSd need to be added to the results of the deployment discipline.
- Additional interfaces: there need to be close links to the disciplines of requirements and business modeling; the diagrams of StSd may serve as boundary objects and be shared among these disciplines.

Exceeding these concrete add-ons to the deployment discipline, the project team would have to revise the assumptions that are identified as problematic with respect to organizational change in Table 11. Most importantly the concept of the bipartite project (cf. Figure 59 and following recommendations) in which the members of the organization, rather than an external design team, are responsible for planning and establishing new work procedures needs to be adopted.

8.3.2 StSd and MUST

MUST is a "*conceptual framework and a coherent method for design in an organizational context within the participatory design tradition*" [Kensing, Simonsen, Bødker (1998), p. 167]; it is introduced and discussed in relation to other methods in the field of CSCW in section 4.3.3 (p. 115). Table 7 summarizes the five activities and the six principles that constitute MUST as a method.

With regard to the six principles, there is a good match between MUST and the concept of StSd; especially the principle of "*co-development of IT, work-organization and users' qualification*" [ibid. pp. 180] coincides with my definition of CSCW-projects (cf. Term 23). Differences can be found in the degree of detail in which the work-organization is actually planned during the project – based on the description in [ibid. pp. 180] my concept of StSd would support a more detailed discussion and yield a more detailed descriptions – and in the way in which documents are used throughout the project – MUST does not foresee to have a guiding artifact.

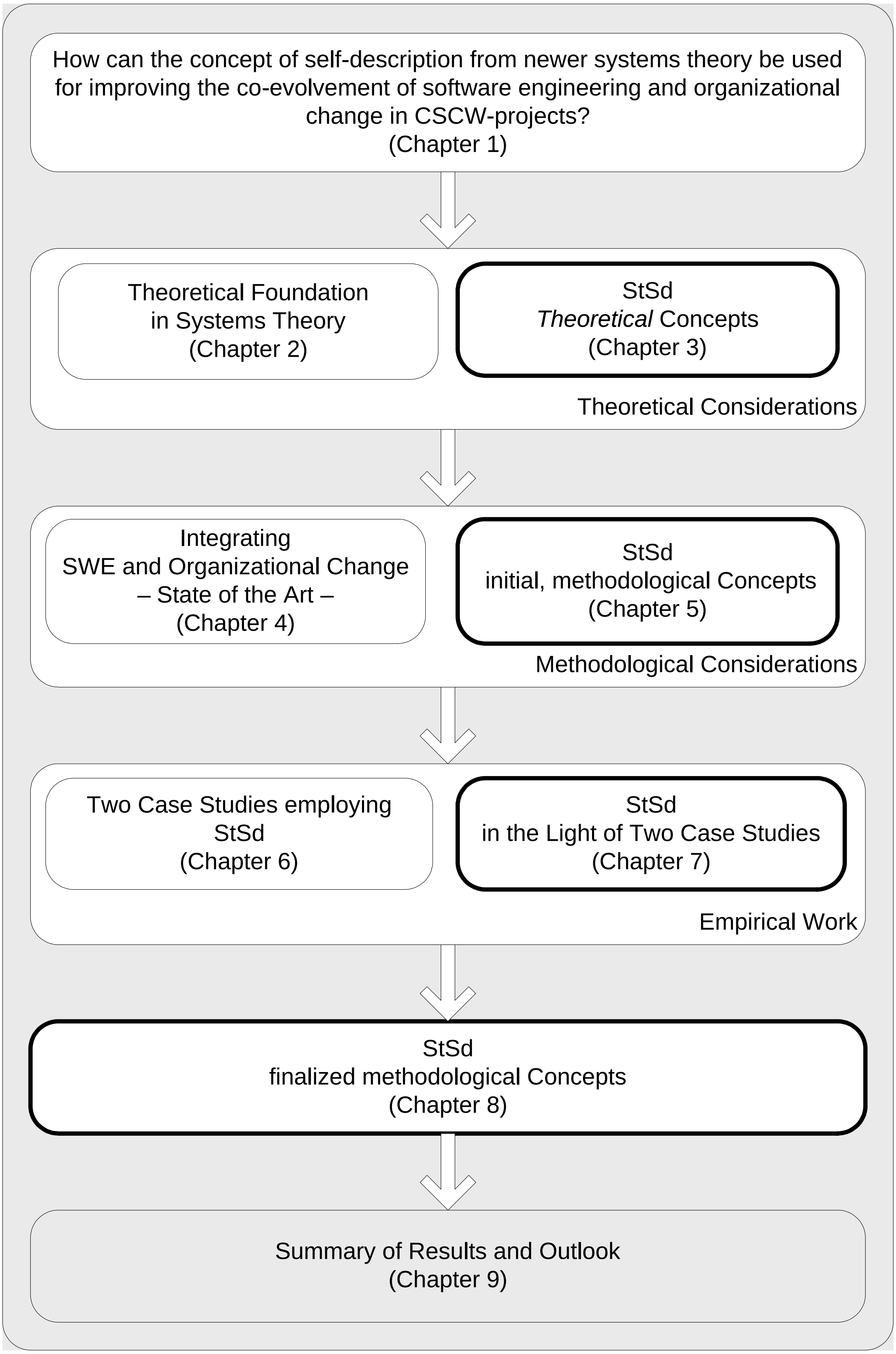
The major difference between my approach of supporting StSd and MUST however, lies in the coverage of project phases. MUST is dedicated to early phases; while StSd are explicitly designed for supporting a complete CSCW-project and reach into the phase of use. But this difference does not hinder the combination of StSd and MUST as far as MUST reaches. In the following it is described how the support of StSd according to the recommendations given in section 8.2 can be integrated into the five processes included in the MUST method.

If StSd are included into MUST, then the idea of maintaining a guiding artifact throughout the whole project is newly introduced into MUST. This affects all of the five activities that will now be discussed in turn. First, the activity of "*Project Establishment*" that sets off a MUST project. This activity includes the negotiation and selection of tools and techniques to be used throughout the project; so here the decision for supporting StSd and for using the adequate tool support needs to be made. Initial diagrams could be used for identifying the scope of the project. The second activity is the "*Strategic analysis*". Although it is described as "*a management related activity*" [ibid. p. 187] diagrams could be used for documenting the results of interviews and analysis; and they could serve as boundary objects between different groups involved in this activity. The third and fourth phase ("*In-Depth Analysis of Selected Work Domains*" and "*Developing Visions for the Overall Change*") are those in which my concept of supporting StSd can contribute the most. Here STWT workshops can be held to complement the workshops already foreseen. I suggest replacing rich pictures and the descriptions on large sheets of paper by semi-structured diagrams in electronic form, which are created and maintained according to the recommendations given in section 8.2. This way StSd can be established as guiding artifacts for both activities and for the project phases that follow design and that are not covered by MUST anymore. Furthermore, StSd as guiding artifacts can be beneficial for the last activity called "*Anchoring Visions*". In this phase which is described as orthogonal to the others, it is important to integrate different stakeholders. By functioning as boundary objects diagrams documenting the StSd can support this activity. The authors of MUST foresee that the activities of "*Anchoring Visions*" also initiate processes of organizational change. Supporting the creation and maintenance of StSd would also support this concern.

8.3.3 Further Options for Integration

There are further options for combining the support for socio-technical self-descriptions with proven methods or techniques that have been discussed within this thesis. These include:

- StSd can be used one tool in Eason's toolbox for socio-technical design of information systems (cf. section 4.3.3, p. 112).
- The support of StSd can be viewed as another methodological component for participatory design (cf. section 4.3.3, p. 113), because it actively involves users and designers.
- The support of StSd can be used to strengthen the methodological repertoire gathered in Multiview II (cf. section 4.3.3, p. 119) with respect to organizational change.
- The support of StSd could be embedded into the STEPS model (cf. section 4.3.3, p. 121) by extending the *embedment preparation* and linking it more closely to *software realization*.
- Depending on the problems that exist in an organization or come up during a CSCW-project, intervention designs from systemic intervention (cf. section 4.2) can be incorporated into the series of STWT-workshops.


How can the concept of self-description from newer systems theory be used for improving the co-evolvement of software engineering and organizational change in CSCW-projects? (Chapter 1)
Theoretical Foundation in Systems Theory (Chapter 2)
StSd *Theoretical* Concepts (Chapter 3)
Theoretical Considerations
Integrating SWE and Organizational Change – State of the Art – (Chapter 4)
StSd initial, methodological Concepts (Chapter 5)
Methodological Considerations
Two Case Studies employing StSd (Chapter 6)
StSd in the Light of Two Case Studies (Chapter 7)
Empirical Work
StSd finalized methodological Concepts (Chapter 8)
Summary of Results and Outlook (Chapter 9)

9 Summary of Results and Outlook

9.1 Summary of Results

Initial Motivation for this Thesis

This thesis started out by observing a special dual characteristic of CSCW-projects: on the one hand they create a technical product which is constructed according to professional engineering methods; on the other they need to establish corresponding work procedures that are brought forth by processes of organizational change. From literature the problem of failing CSCW-projects was identified and the downside of the phenomenon of evolving use was discussed. As one source for problems, a gap in the methodological support for CSCW-projects was identified: methodological components were missing for supporting the process of structural coupling in parallel with the process of software engineering.

The concept of self-description from Luhmann's newer system theory was introduced, and the question to be answered by this thesis was phrased: "How can the concept of self-description from newer system theory be used for improving the co-evolvement of software engineering and organizational change in CSCW-projects?"

Framework for Socio-technical Self-descriptions

In answering this question, I elaborated a framework for socio-technical self-descriptions comprising:

1. The theoretical concepts describing interdependencies between an organization and a CSCW-system: socio-technical setting, socio-technical organization and socio-technical self-description (cf. chapter 3).
2. A process model for CSCW-projects that integrates the processes of organizational change and technical software engineering by supporting the creation and maintenance of StSd (cf. Figure 59).
3. A set of seven recommendations providing methodological support for documenting socio-technical self-descriptions in CSCW-projects that can be integrated with established methods in the fields of software engineering and CSCW (cf. chapter 8).

The framework proved its worth in two case studies and can be applied within CSCW-projects in combination with different project models.

While the framework for socio-technical self-descriptions is the central and most important result of my work, there are other results which are also worth mentioning.

Design Implications for Software Systems

The empirical work in the case studies allowed the identification of design implications for two types of software systems: for editors which are used for the semi-structured diagrams, and for CSCW-systems themselves.

With regard to the editor I identified four settings for which it should offer dedicated support (cf. section 7.3.1, pp. 234):

- Co-located collaboration: support for large interactive screens; support for rapid drawing and sketching; markup of changes.
- Individual editing: semi-automated support for "aesthetical improvement" of diagrams; support for effective creation and change of hyperlinks to additional material; integration of diagrams and additional material into one repository; semi-automated support for preparation of presentations.
- Presentations: support for integration with other forms of presentations; support for automatically adjusting the size of the overall diagrams depending on the number of visible elements ("shrinking").
- Individual Reading: support for semi-automated presentations; provision of an interface that allows CSCW-applications to link into specific parts of a diagram and to start the semi-automated reader.

If the methodological concept for StSd is followed, then it is desirable that the CSCW-system, which contains the inscribed part of an organization's self-description, were able to relate to the other two forms of socio-technical self-descriptions: ephemeral and documented. With regard to the CSCW-systems the following design implications can be derived from this idea (cf. section 7.2.2, pp. 220):

- CSCW-systems should provide an interface that allows the insertion of context-sensitive links to documented forms of socio-technical self-descriptions.
- CSCW-systems should include communication channels that allow users to engage in a meta-communication about the usage of the CSCW-system; this supports the users in creating and maintaining socio-technical self-descriptions during the phase of use.

These recommendations apply to different types of CSCW-systems such as group calendars, workflow systems, shared information spaces etc. (cf. section 2.2.4).

Theoretical Conceptualizations for "socio-technical Systems"

So far the theoretical base elaborated from Luhmann's system theory and included in the framework for StSd has been viewed as an integral part of this framework. However chapters 2 and 3 contain an abundant source of theoretical concepts that may prove to be useful in other areas of computer science as well.

Chapter 2 contains an introduction into system theory that continuously explicates its explanatory potential for CSCW-projects. Chapter 3 contains a reconceptualization of "socio-technical systems" that exceeds existing definitions with respect to its explanatory potential as well as with respect to its potential for being the base of a methodological concept.

Two ways in which other areas of computer science could make use of this theoretical base are:

- The clarification of the relation between software engineering and organizational change, and the description of interfaces between the two. Section 4.1.8 discussed problematic aspects of software engineering concepts that extend their methodological repertoire into the field of organizational change. The interdependence of software engineering and organizational change is generally acknowledged for cooperative software systems, but a concept that is equally acknowledged and that describes how the complementary methods should be integrated is missing.
- Artificial intelligence, the design of multiagent systems: the concepts by which system theory explains how autopoietically closed systems such as psychic systems or organizations succeed in interacting with their environment could inform the design of multiagent systems. It would be a perspective adding to concepts such as game theory.

9.2 Outlook

The research presented in this thesis lays the ground for further research; a few strands shall be outlined.

The framework for supporting StSd is now applied without my direct involvement. For the development and configuration of a CSCW-system in a large medical practice STWT workshops were conducted and StSd were created using SeeMe-Diagrams. The project leader consulted me to advise him how to conduct the project, but no researcher was involved in the actual project work. This will be a chance to evaluate whether the concepts are indeed transferable to other projects and other project teams. More projects of this kind will offer the possibility for initiating research projects other than action research.

In case study 1 the documents created for the purposes of socio-technical self-description were mostly kept apart from those necessary for software engineering. This helped to identify the indispensable characteristics of socio-technical self-description. In case study 2, some aspects of requirements engineering and even database design were included into the socio-technical self-descriptions. It would be interesting to evaluate how far socio-technical self-descriptions can be extended into the process of technical software design and implementation without loosing their identity as socio-technical self-descriptions. In this context semi-automated functions for converting socio-technical self-descriptions into more technical documents such as workflow descriptions would be feasible.

It would be interesting to initiate projects for realizing the design implications identified for editors as well as for CSCW-systems. The resulting software systems should be used in CSCW-projects again; the experience should then inform the further development of the framework to include the advanced technical support.

In section 2.2.4 KBS were mentioned as one type of software systems that can be used in cooperative work settings. Until now, the field of CSCW has not paid much attention to this kind of software system. It would be interesting to analyze the potential of KBS for cooperative work more closely and to apply the concept of socio-technical self-description to the design, deployment and use of a KBS. One question would be, whether the contents of StSd I identified in this thesis would change if a KBS is designed.

I suggested that diagrams for socio-technical self-descriptions are discussed using large interactive displays supporting co-located collaboration. It would be interesting to regard groups involved in technically supported, co-located collaboration together with the corresponding technical systems as socio-technical settings again and to derive methodological support for this special socio-technical setting.

10 Bibliography and Lists

10.1 Print

Andriessen, J.H. Eric (2003): Working with groupware: understanding and evaluating collaboration technology. London: Springer.

Andriessen, J.H. Eric; Hettinga, Mareike; Wulf, Volker (2003): Introduction to Special Issue on Evolving Use of Groupware. In: Computer Supported Cooperative Work 12. pp. 367 - 380.

Atkinson, Christopher J. (2000): Socio-Technical and Soft Approaches to Information Requirements Elicitation in the Post-Methodology Era. In: Requirements Engineering 5. pp. 67 - 73.

Atkinson, Christopher J. (2000a): The 'Soft Information Systems and Technologies Methodology' (SISTeM): an actor network contingency approach to integrated development. In: European Journal of Information Systems 9. pp. 104 - 123.

Avison, D.E.; Wood-Harper, A.T.; Vidgen, R.T.; Wood, J.R.G. (1998): A further exploration into information systems development: the evolution of Multiview2. In: Information Technology & People Vol. 11, No. 2. pp. 124 - 139.

Avison, David E.; Wood-Harper, A.T. (1990): Multiview: An Exploration in Information Systems Development. Oxford: Blackwell Scientific Publications.

Baecker, Dirk (2002): Wozu Systeme?. Berlin: Kulturverlag Kadmos.

Beck, Kent (2004): Extreme Programming Explained. Boston: Addison-Wesley.

Beckhard, Richard (1969): Organization Development: Strategies and Models. Reading, Massachusetts: Addison-Wesley Publishing Company.

Beierle, Christoph; Kern-Isberner, Gabriele (2000): Methoden wissensbasierter Systeme. Braunschweig, Wiesbaden: Vieweg.

Bennis, Warren G. (1969): Organization Development: ist Nature, Origins, and Prospects. Reading, Massachusetts: Addison-Wesley Publishing Company.

Berning, Markus (2004): Dynamische und interaktive Präsentation graphischer Prozessdiagramme. Master Thesis, Universität Dortmund.

Beyer, Hugh; Holtzblatt, Karen (1998): Contextual Design: Defining Customer-Centered Systems. San Francisco: Morgan Kaufmann Publ.

Birkelbach, Jörg; Schulzki-Haddouti, Christiane (2004): Amt im Netz. In: c't Heft 8. pp. 158 - 162.

BMBF, Bundesministerium für Bildung und Forschung (2002): Innovative development of work - The future of work. Bonn: BMBF, Referat Öffentlichkeitsarbeit.

Boehm, Barry W. (1988): A Spiral Model of Software Development and Enhancement. In: IEEE Computer May 1988. pp. 61 - 72.

Boos, Frank; Jarmai, Heinz (2004): "Reengineering mit offenen Karten" Gemeinsam mit den Mitarbeitern für oder gegen die Mitarbeiter. In: Königswieser, Roswita; Exner, Alexander (Eds.): Systemische Intervention - Architekturen und Designs für Berater und Veränderungsmanager. Stuttgart: Klett-Cotta.

Brachman, Ronald J.; Levesque, Hector J. (2004): Knowledge Representation and Reasoning. San Francisco: Elsevier.

Broy, Manfred; Rausch, Andreas (2005): Das neue V-Modell XT - Ein anpassbares Modell für Software und System Engineering. In: Informatik Spektrum Band 22. pp. 220 - 229.

Buckland, Michael K. (1997): What is a document? In: Journal of the American Society of Information Science Volume 48, No. 9. pp. 804 - 809.

Carell, Angela; Herrmann, Thomas; Kienle, Andrea; Menold, Natalja (2005): Improving the Coordination of Collaborative Learning with Process Models. In: Koschmann, Tim; Suthers, Dan; Chan, T.W. (Eds.): Proceedings of CSCL 2005. The Next 10 Years. Mahwah, New Jersey: LEA. pp. 18 - 27.

Carmel, Erran; Whitaker, Randall D.; George, Joey F. (1993): PD and Joint Application Design: A Transatlantic Comparison. In: Communications of the ACM Vol 36, No. 4. pp. 40 - 48.

Carroll, John M. (1995): The Scenario Perspective on System Development. In: Carroll, John M. (Ed.): Scenario Based Design: Envisioning Work and Technology in System Development. New York: Wiley & Sons. pp. 1 - 17.

Carroll, John M. (1997): Becoming social. Expanding scenario-based approaches in HCI. In: Behaviour & Information Technology Vol. 15, No. 4. pp. 266 - 275.

Carroll, John M. (2000): Making Use. Cambridge: MIT Press.

Checkland, Peter (1981): Systems Thinking, Systems Practice. Chichester: John Wiley & Sons.

Checkland, Peter (1999): Systems Thinking, Systems Practice. Chichester, England: John Wiley & Sons, LTD.

Checkland, Peter; Scholes, Jim (1990): Soft Systems Methodology in Action. Chichester: John Wiley & Sons.

Checkland, Peter; Holwell, Sue (1998): Information, Systems and Information Systems. Chichester: John Wiley & Sons.

Cherns, Albert (1987): Principles of sociotechnical Design revisited. In: Human Relations Volume 40. pp. 153 - 161.

Coakes, Elayne; Willis, Dianne; Lloyd-Jones, Raymond (2000): Graffiti on the Long Wall: A SocioTechnical Conversation. In: Coakes, Elayne; Willis, Dianne; Lloyd-Jones, Raymond (Eds.): The New SocioTech. Great Britain: Springer-Verlag London Limited. pp. 3 - 12.

Cockburn, Alistair (2001): Writing effective use cases. Addison-Wesley.

Cockburn, Alistair (2002): Agile Software Development. Addison-Wesley.

Cockton, Gilbert; Lavery, Darry; Woolrych, Alan (2003): Inspection-Based Evaluations. In: Jacko, Julie A.; Sears, Andrew (Eds.): The Human-Computer Interaction Handbook. Mahwah, New Jersey: LEA. pp. 1118 - 1138.

Coughlan, Jane; Macredie, Robert D. (2002): Effective Communication in Requirements Elicitation: A Comparison of Methodologies. In: Requirements Engineering 7. pp. 47 - 60.

Crabtree, Andy; Hemmings, Terry; Rodden, Tom; Mariani, John (2003): Informing the Development of Calendar Systems for Domestic Use. In: Kuutti, Kaari; Karsten, Elena H.; Fitzpatrick, Geraldine; Dourish, Paul; Schmidt, Kjeld (Eds.): Proceedings of the Eighth European Conference on Computer Supported Cooperative Work. Kluwer Academic Publishers. pp. 119 - 138.

Crawford, Anthony (1994): Advancing Business Concepts in a JAD Workshop Setting. New Jersey: Yourdon Press.

Cummings, Thomas G. ; Srivastva, Suresh (1977): Management of Work - A socio-technical Systems Approach. Kent, Ohio: Kent State University Press.

DeSanctis, G.; Poole, M.S. (1994): Capturing the Complexity in Advanced Technology Use: Adaptive Structuration Theory. In: Organization Science Volume 5, No. 2. pp. 121 - 147.

Dixit, Avinash; Skeath, Susan (2004): Games of Strategy. New York, London: W.W. Norton & Company.

Dourish, Paul (2001a): Process Descriptions as Organisational Accounting Devices: The Dual Use of Workflow Technologies. In: Rodden, Tom; Ellis, Clarence; Zigurs, Ilze (Eds.): Proceedings of GROUP'01 Conference on Supporting Group Work. New York, NY, USA: ACM Press. pp. 52 - 60.

Dourish, Paul (2001b): Where the action is. Cambridge, Massachusetts: MIT Press.

Dourish, Paul (2003): The appropriation of interactive technologies: some lessons from Placeless Documents. In: Computer Supported Cooperative Work 12. pp. 465 - 490.

Dunckel, Heiner (2003): Leitfaden zur Kontrastiven Aufgabenanalyse (Kaba). In: Dunckel, Heiner (Ed.): Handbuch psychologischer Arbeitsanalyseverfahren. Zürich: vdf Hochschulverlag. pp. 231 - 254.

Eason, Ken (1988): Information Technology and Organisational Change. London, New York, Philadelphia: Taylor & Francis.

Ehn, Pelle; Sjögren, Dan (1991): From System Descriptions to Scripts for Action. In: Greenbaum, Joan; Kyng, Morten (Eds.): Design at Work: Cooperative Design of Computer Systems. Hilldale, NJ: Lawrence Erlbaum. pp. 168 - 195.

Emery, Fred E. (1993): The Nine-Step Model. In: Trist, Eric; Murray, Hugh (Eds.): The social engagement of social science - A Tavistock Anthology. Philadelphia: University of Pennsylvania Press. pp. 569 - 579.

Emery, Fred E.; Thorsrud, Einar (1976): Democracy at work. Leiden: Martinus Nijhoff Social Sciences Division.

Emery, Fred E.; Trist, Eric L. (1960): Socio-technical Systems. In: Churchman, C. West; Verhulst, Michel (Eds.): Management Sciences Models and Techniques. Proceedings of the sixth international Meeting of the Institute of Management Science. Oxford, London, New York, Paris: Pergamon Press. pp. 83 - 97.

Emery, Fred E.; Trist, Eric L. (1969): Socio-technical Systems. In: Emery, Fred E. (Ed.): System Thinking. Harmondsworth: Penguin. pp. 281 - 296.

Endres, Albert (2003): Professionalität und Verantwortung in der Informatik. In: Informatik-Spektrum Aug 03. pp. 261 - 266.

Engeström, Yrjö (2000): From individual action to collective activity and back: developmental work research as an interventionist methodology. In: Luff, Paul; Hindmarsh, Jon; Heath, Christian (Eds.): Workplace Studies - Recovering Work Practice and Informing System Design. Cambridge: University Press. pp. 150 - 166.

Engeström, Yrjö; Miettinen, Reijo (1999): Introduction. In: Engeström, Yrjö; Miettinen, Reijo; Punamäki, Raija-Leena (Eds.): Perspectives on Activity Theory. Cambridge: University Press. pp. 1 - 16.

Erkens, Elmar; Kopfer, Herbert; Siek, Katja (2005): Prozessanalyse und Prozessmodell. In: Herrmann, Thomas; Schöpe, Lothar; Erkens, Elmar; Hülder, Malte (Eds.): Mobile Speditionslogistikunterstützung. Aachen: Shaker Verlag. pp. 3-1- 3-21.

Fank, Matthias (1998): Tools zur Geschäftsprozeßorganisation. Braunschweig, Wiesbaden: Vieweg.

Fischer, G.; Giaccardi, E.; Ye, Y.; Sutcliffe, A.G.; Mehandjiev, N. (2004): Meta-design: a manifesto for end-user development. In: Communications of the ACM Volume 7, No. 9. pp. 33 - 37.

Floyd, Christiane; Krabbel, Anita; Ratuski, Sabine; Wetzel, Ingrid (1997): Zur Evolution der evolutionären Systementwicklung: Erfahrungen aus einem Krankenhaus. In: Informatik Spektrum 20. Heidelberg: Springer Verlag. pp. 13 - 20.

Floyd, Christiane; Reisin, F.-M.; Schmidt, G. (1989): STEPS to Software Development with Users. In: Ghezzi, J.; McDermid, J. A. (Eds.): Lecture Notes in Computer Science Nr. 387. Heidelberg: Springer Verlag. pp. 48 - 64.

Floyd, Christiane; Züllighoven, H. (2002): Softwaretechnik. In: Rechenberg, Peter; Blaschek, Günther (Eds.): Informatik-Handbuch. München: Hanser. pp. 764 - 790.

French, Wendell L.; Bell, Cecil H. (1973): Organization Development - behavioral science interventions for organization improvement. Englewood Cliffs, New Jersey: Prentice Hall.

Galliers, R.D.; Swan, J.A. (2000): There's More to Information Systems Development than Structured Approaches: Information Requirements Analysis as a Socially Mediated Process. In: Requirements Engineering 5. pp. 74 - 82.

Gebert, Diether; Gaawlik, Rainer; Herzig, Hans Ulrich (1974): Organisationsentwicklung - Probleme des geplanten organisatorischen Wandels (Sozioökonomie 6). Stuttgart, Berlin, Köln, Mainz: Verlag W. Kohlhammer.

Gerson, E.M.; Star, Susan Leigh (1986): Analyzing Due Process in the Workplace. In: ACM Transactions on Office Information Systems Volume 4, No. 3. pp. 257 - 270.

Giddens, Anthony (1984): The Constitution of Society. Berkeley: University of California Press.

Götz, Klaus (1998): Theoretische Zumutungen. Vom Nutzen der systemischen Theorie für die Managementpraxis. Heidelberg: Carl-Auer-Systeme Verlag.

Greenbaum, Joan; Kyng, Morten (Eds.) (1991): Design at Work: Cooperative Design of Computer Systems. Hillsdale, New Jersey: Lawrence Erlbaum Associates Inc. Pub..

Grudin, Jonathan (1988): Why CSCW applications fail: problems in the design and evaluation of organization of organizational interfaces. In: Proceedings of the conference on Computer-supported cooperative work, September 26 - 28, 1988, Portland, OR. pp. 85 - 93.

Gruhn, Volker; Hülder, Malte; Ijioui, R; Schleif, Frank-Michael; Schöpe, Lothar (2003): A Distributed Logistic Support Communication System. In: Linger, H.; Fisher, J.; Wojtkowski, W.G.; Zupancic, J.; Vigo, K.; Arnold, J. (Eds.): Constructing the Infrastructure for the Knowledge Economy - Methods and Tools, Theory and Practice. Dordrecht, Boston, London: Kluwer Academic Publishers. pp. 705 - 713.

Gutwin, Carl; Greenberg, Saul (2002): A Descriptive Framework of Workspace Awareness for Real-Time Groupware. In: Computer Supported Cooperative Work Volume 11. pp. 411 - 446.

Haase, Christoph (2003): Konzeption und Realisierung einer verteilten Workflow-Steuerung für mobile Endgeräte. Universität Dortmund.

Hammer, Michael; Champy, James (1994): . In: Reengineering the Corporation. USA: Harper Collins Publishers, Inc..

Heinrich, Lutz J. (2002): Grundlagen der Wirtschaftsinformatik. In: Rechenberg, Peter; Pomberger, Gustav (Eds.): Informatik-Handbuch. München, Wien: Hanser. pp. 1039 - 1054.

Henderson, Austin; Kyng, Morten (1991): There's No Place Like Home: Continuing Design in Use. In: Greenbaum, Joan; Kyng, Morten (Eds.): Design at Work: Cooperative Design of Computer Systems. Hillsdale, New Jersey: Lawrence Erlbaum Associates Inc. Pub.. pp. 219 - 240.

Herrmann, Thomas (1997): Communicable Models for Cooperative Processes. In: Slavendy, G. (Ed.): Proceedings of the 7th International Conference on Human-Computer Interaction, San Francisco. Amsterdam: Elsevier. pp. 285 - 288.

Herrmann, Thomas (1999a): Kompendium zur Grundvorlesung "Informatik und Gesellschaft".

Herrmann, Thomas; Hoffmann, Marcel (in print): The Metamorphoses of Workflow Projects in their Early Stages. In: Computer Supported Cooperative Work.

Herrmann, Thomas; Hoffmann, Marcel; Kunau, Gabriele; Loser, Kai-Uwe (2002a): Modelling Cooperative Work: Chances and Risks of Structuring. In: Blay-Fornarino, M.; Pinna-Dery, Anne Marie; Schmidt, Kjeld; Zaraté, Pascale (Eds.): Cooperative Systems Design, A Challenge of the Mobility Age (Coop 2002). Amsterdam: IOS Press. pp. 53 - 70.

Herrmann, Thomas; Hoffmann, Marcel; Kunau, Gabriele; Loser, Kai-Uwe (2004a): A Modeling Method for the Development of Groupware Applications as Socio-Technical Systems. In: Behaviour & Information Technology Volume 23, No. 2. pp. 119 - 135.

Herrmann, Thomas; Hoffmann, Marcel; Loser, Kai-Uwe; Moysich, Klaus (2000): Semistructured models are surprisingly useful for user-centered design. In: Dieng, R. ; Giboin, A.; Karsenty, L.; DeMichelis, George (Eds.): Designing Cooperative Systems. Proceedings of Coop 2000. Amsterdam: IOC press. pp. 159 - 174.

Herrmann, Thomas; Kunau, Gabriele; Loser, Kai-Uwe (2002b): Sociotechnical Walkthrough - ein methodischer Beitrag zur Gestaltung soziotechnischer Systeme. In: Herczeg, Michael; Prinz, Wolfgang; Oberquelle, Horst (Eds.): Mensch & Computer. Vom interaktiven Werkzeug zu kooperativen Arbeits- und Lernwelten. Stuttgart: Teubner. pp. 323 - 332.

Herrmann, Thomas; Kunau, Gabriele; Loser, Kai-Uwe; Menold, Natalja (2004b): Sociotechnical Walkthrough: Designing Technology along Work Processes. In: Clement, Andrew; Cindio, Fiorella; Oostveen, Anne-Marie; Schuler, Doug; van den Besselaar, Peter (Eds.): Artful Integration: Interweaving Media, Materials and Practices. Proceedings of the eighth Participatory Design Conference 2004. New York: ACM Press. pp. 132 - 414.

Herrmann, Thomas; Loser, Kai-Uwe (1999): Vagueness in models of socio-technical systems. In: Behaviour & Information Technology Volume 18, No. 5. pp. 313 - 323.

Hill, Paul (1971): Towards a new philosophy of management. Epping, Essex: Gower Press Limited.

Ho, Thao (2004): Modellgestützte Moderation und Dokumentation eines soziotechnischen Entwicklungszyklus am Beispiel von Bibliotheksaufgaben. Master Thesis, Universität Dortmund.

Holtzblatt, Karen (2003): Contextual Design. In: Jacko, Julie A. ; Sears, Andrew (Eds.): The Human-Computer Interaction Handbook. Mahwah, New Jersey: LEA. pp. 941 - 963.

Holtzblatt, Karen; Beyer, Hugh (1996): Contextual Design: Principles and Practice. In: Wixon, Dennis; Ramey, Judith (Eds.): Field Methods Casebook for Software Design. New York: John Wiley & Sons, Inc.. pp. 301 - 333.

Huber, Andreas ; Kuhnt, Beate (1995): A Systemic Approach to Application Design. In: Technical Report.Universität Zürich, Institut für Informatik.

Hult, Margareta; Lennung, Sven-Ake (1980): Towards a definition of action research: a note and bibliography. In: Journal of Management Studies Volume 17. pp. 241 - 250.

Jacobson, Ivar (1993): Object oriented software engineering. A use case driven approach. Wokingham: Addison-Wesley.

Jeckle, Mario; Rupp, Chris; Hahn, Jürgen; Zengler, Barbara; Queins, Stefan (2004): UML2 glasklar. München Wien: Carl Hanser Verlag.

Joas, Hans; Knöbl, Wolfgang (2004): Sozial Theorie - Zwanzig einführende Vorlesungen. Frankfurt am Main: Suhrkamp Taschenbuch Wissenschaft.

Jones, Matthew (1999): Structuration Theory. In: Currie, Wendy L.; Galliers, Bob (Eds.): Rethinking Management Information Systems. Oxford: University Press. pp. 103 - 135.

Katzenstein, Gary; Lerch, F. Javier (2000): Beneath the Surface of Organizational Processes: A Social Representation Framework for Business Process Redesign. In: ACM Transactions on Information Systems Volume 18, No. 4. pp. 383 - 422.

Kensing, Finn; Blomberg, Jeanette (1998): Participatory Design: Issues and Concerns. In: Computer Supported Cooperative Work 7. pp. 167 - 185.

Kensing, Finn; Munk-Madsen, Andreas (1993): PD: Structure in the Toolbox. In: Communications of the ACM Voume 36, No. 4. pp. 78 - 85.

Kensing, Finn; Simonsen, Jesper; Boedker, Keld (1998): MUST: A Method for Participatory Design. In: Human-Computer Interaction 13. pp. 167-198.

Kienle, Andrea (2002): . In: Integration von Wissensmanagement und kollaborativem Lernen durch technisch unterstützte Kommunikationsprozesse. Lohmar: Josef Eul Verlag.

Klebert, Karin; Schrader, Einhard; Straub, Walter G. (1987): Kurzmoderation. Hamburg: Windmühle.

Klepper, Robert ; Hoffmann, Norton (2000): Assimilation of new information technology and organizational culture: a case study. In: Wirtschaftsinformatik Heft 4. pp. 339 - 346.

Klink, Markus; Oestereich, Bern (2005): Das neue V-Modell XT. In: OBJEKTspektrum No. 4 (Juli/August). pp. 55 - 57.

Klischewski, Ralf (2002): Reaching out for Commitments: System Development as Networking. In: Dittrich, Yvonne; Floyd, Christiane; Klischewski, Ralf (Eds.): Social Thinking - Software Practice. Cambridge, Massachusetts: MIT Press. pp. 309 - 329.

Kneer, Georg; Nassehi, Armin (2000): Niklas Luhmanns Theorie sozialer Systeme (4. Auflage). München: Werner Fink Verlag.

Königswieser, Roswita; Exner, Alexander (2004): Systemische Intervention - Architekturen und Designs für Berater und Veränderungsmanager. Stuttgart : Klett-Cotta.

Krcmar, Helmut (2005): Informationsmanagement. Springer Verlag.

Kruchten, Philippe (2004): The Rational Unified Process: An Introduction (Third Edition). Boston: Addison-Wesley.

Kuhnt, Beate (1997): Systemische Beratung in kooperativen Softwareprojekten. In: Informatik-Spektrum 1. pp. 27 - 33.

Kuhnt, Beate (1998): Softwareentwicklung als systemische Intervention in Organisationen. Zürich: Dissertation der wirtschaftswissenschaftlichen Fakultät der Universität Zürich.

Kuutti, Kari (1996): Debates in IS and CSCW Research: Anticipating System Design for Post-Fordist Work. In: Orlikowski, Wanda J.; Walsham, Geoff; Jones, Matthew R. (Eds.): Information Technology and Changes in Organizational Work. London: Chapman & Hall. pp. 176 - 196.

Langer, Thorsten (2005): Konsistenz und Persistenz von SeeMe-Modellen durch Metamodellierung. Master Thesis, Universität Dortmund.

Latour, Bruno (1998): Über technische Vermittlung. In: Rammert, Werner (Ed.): Technik und Sozialtheorie. Frankfurt / New York: Campus Verlag. pp. 29 - 82.

Loser, Kai-Uwe; Herrmann, Thomas (2001): Diagrammatic Models for Participatory Design of Socio-Technical Systems. In: Bjornestad, S.; Moe, R. ; Morch, Anders; Opdahl, A. (Eds.): IRIS 24. Proceedings of the 24th Information Systems Research Seminar in Scandinavia. pp. 285 - 300.

Loser, Kai-Uwe; Herrmann, Thomas (2002): Enabling factors for participatory design of socio-technical systems with diagrams. In: Binder, T.; Gregory, J.; Wagner, Ina (Eds.): PDC 02 - Proceedings of the Participatory Design Conference. Palo Alto: CPSR.

Luczak, H.; Wimmer, R.; Schuhmann, R. (1996): Rechnergestütztes Verfahren zur Modellierung und handlungsregulatorischen Bewertung von Arbeitstätigkeiten. In: Zeitschrift für Arbeitswissenschaft 50 (NF 22). pp. 70 - 79.

Luff, Paul; Hindmarsh, J; Heath, Christian (Eds.) (2000): Workplace Studies - Recovering Work Practice and Informing System Design. Cambridge: University Press.

Luhmann, Niklas (1984): Soziale Systeme Grundriß einer allgemeinen Theorie. Frankfurt a.M.: Suhrkamp Verlag.

Luhmann, Niklas (1986): The autopoiesis of social systems. In: Geyer, R. Felix; Zouwen, Johannes van der (Eds.): Sociocybernetics: observation, control and evolution of self steering systems. London: Sage Publications Limited.

Luhmann, Niklas (1990): Essays on Self-Reference. New York: Columbia University Press.

Luhmann, Niklas (1995): Social Systems (Translated by Bednarz Jr., John with Baecker, Dirk). Stanford, California: Stanford University Press..

Luhmann, Niklas (2000): Organisation und Entscheidung. Opladen/Wiesbaden: Westdeutscher Verlag.

Luhmann, Niklas (2002): Einführung in die Systemtheorie. Heidelberg: Carl-Auer-Systeme Verlag.

Mark, Gloria (2002): Conventions and Commitments in Distributed CSCW Groups. In: Computer Supported Cooperative Work Volume 11. pp. 349 - 387.

Maturana, Humberto; Varela, Francisco (1987): The Tree of Knowledge. Boston, MA: New Science Library.

McKay, Judy; Marschall, Peter (2001): The dual imperatives of action research. In: Information Technology & People Vol. 14 No. 1. pp. 46 - 59.

McKenzie, Adrian; Monk, Simon (2004): From Cards to Code: How Extreme Programming Re-Embodies Programming as a Collective Practice. In: Computer Supported Cooperative Work Volume 13, No. 1. pp. 91 - 117.

Menold, Natalja (Manuscript in preparation): Wissensintegration und Handeln in Gruppen: Unterstützung bei der Gestaltung sozio-technischer Systeme.

Mittelstraß, Jürgen (1998): Die Häuser des Wissens. Frankfurt am Main: Suhrkamp.

Moll, Karl-Rudolf; Broy, Manfred; Pizka, Markus; Seifert, Tilman; Bergner, Klaus; Rausch, Andreas (2004): Erfolgreiches Management von Software-Projekten. In: Informatik Spektrum Band 27, Heft 5. pp. 419 - 432.

Monteiro, Eric; Hanseth, Ole (1996): Social Shaping of Information Infrastructure: On Being Specific about the Technology. In: Orlikowski, Wanda J.; Walsham, Geoff; Jones, Matthew R. (Eds.): Information Technology and Changes in Organizational Work. London: Chapman & Hall. pp. 325 - 343.

Muller, Michael J.; Wildman, Daniel M.; White, Ellen A. (1993): Taxonomy of PD Practices: A Brief Practioner's Guide. In: Communications of the ACM Volume 36, No. 4. pp. 26 - 27.

Mumford, Enid (1995): Effective Systems Design and Requirements and Analysis - the ETHICS aproach. Houndsmill, Basingstoke, Hampshire and London: Macmillan Press LTD.

Mumford, Enid (1996): Systems Design- Ethical Tools for Ethical Change. Houndsmill, Basingstoke, Hampshire and London: Macmillan Press LTD..

Mumford, Enid (2000): A Socio-Technical Approach to Systems Design. In: Requirements Engineering 5. pp. 125 - 133.

Oberquelle, Horst (1987): Human-Machine Interaction and Role / Function / Action-Nets. In: Brauer, W.; Reisig, W.; Rozenberg, G. (Eds.): Petri Nets: Applications and Relationships to Other Models of Concurrency. Berlin et al.: Springer. pp. 171 - 190.

Oberquelle, Horst (Ed.) (1991): Kooperative Arbeit und Computerunterstützung. Stand und Perspektiven. Göttingen, Stuttgart: Verlag für Angewandte Psychologie.

Orlikowski, Wanda J. (1992a): Learning from Notes: Organisational Issues in Groupware Implementation. In: Proceedings of the CSCW1992. pp. 362 - 369.

Orlikowski, Wanda J. (1992b): The duality of technology: Rethinking the Concept of Technology in Organizations. In: Organization Science Volume 3, No. 3. pp. 398 - 427.

Orlikowski, Wanda J. (1996): Improvising Organizational Transformation Over Time: A Situated Change Perspective. In: Information Systems Research Volume 7, No. 1. pp. 63-92.

Orlikowski, Wanda J.; Gash, Debra C. (1994): Technological Frames: Making Sense of Information Technology in Organizations. In: ACM Transactions on Information Systems Volume 12, No. 2. pp. 174 - 207.

Pankoke-Babatz, Uta (2003): Designkonzept für Systeme zur computergestützten Zusammenarbeit unter Nutzung der Behavior-Setting-Theorie. Aachen: Shaker Verlag.

Parnas, David Lorge; Madey, Jan (1995): Functional documents for computer systems. In: Science of Computer Programming Volume 25. pp. 41 - 61.

PAS 1021, (2003): Verfahrensmodell zur Gestaltung von Geschäftsprozessen der öffentlichen Verwaltung - Wandel von der funktionalen zu prozessorientierten Verwaltung. Berlin: Beuth Verlag.

Pendergast, Mark; Aytes, Kregg; Lee, James D. (1999): Supporting the group creation of formal and informal graphics during business process modeling. In: Interacting with Computers 11 (4). pp. 355 - 373.

Pinelle, David; Gutwin, Carl (2002): Groupware Walkthrough: Adding Context to Groupware Usability Evaluation. In: CHI letters Volume 4, Issue 1. pp. 455 - 462.

Pipek, Volkmar (2005): From Tailoring to Appropriation Support: Negotiating Groupware Usage. Oulu: Oulu University Press.

Polson, P.G.; Lewis, C.; Riemann, J.; Wharton, C. (1992): Cognitive walkthrough: a method for theory-based evaluation of user interfaces. In: Int. J. f. Man-Machine Studies 36. Academic Press. pp. 741 - 773.

Prinz, Wolfgang (1993): TOSCA. Providing organizational information to CSCW applications. In: deMichelis, Giorgio; Simone, Carla; Schmidt, Kjeld (Eds.): Proceedings of the Third European Conference on Computer Supported Cooperative Work 13-17 Sept. 1993, Milan, Italy (ECSCW`93). Dordrecht, Boston, London: Kluwer Academic Publishers. pp. 139 - 154.

Prinz, Wolfgang (1998): Erfahrungen und Empfehlungen aus dem Designprozess einer evolutionären Groupware-Entwicklung. In: Herrmann, Thomas; Just-Hahn, Katharina (Eds.): Groupware und organisatorische Innovation (D-CSCW'98). Stuttgart: B.G. Teubner. pp. 139 - 151.

Prinz, Wolfgang; Mark, Gloria; Pankoke-Babatz, Uta (1998): Designing Groupware for Congruency in Use. In: Proceedings of CSCW'98, Seattle, USA. ACM Press. pp. 373 - 382.

Robinson, Mike (1993a): Design for unanticipated use …. In: deMichelis, Giorgio; Simone, Carla; Schmidt, Kjeld (Eds.): Proceedings of the Third European Conference on Computer Supported Cooperative Work 13-17 Sept. 1993, Milan, Italy (ECSCW`93). Dordrecht, Boston, London: Kluwer Academic Publishers. pp. 187 - 202.

Robinson, Mike (1993b): Keyracks and computers: an introduction to "common artefact" in Computer Supported Cooperative Work (CSCW). In: Wirtschaftsinformatik 35, 2. pp. 157 - 166.

Robinson, Mike; Bannon, Liam (1991): Questioning Representations. In: Bannon, Liam; Robinson, Mike; Schmidt, Kjeld (Eds.): Proceedings of the Second European Conference on Computer Supported Cooperative Work. Dordrecht, Boston, London: Kluwer Academic Publishers. pp. 219 - 233.

Rogers, Yvonne; Lindley, Siân (2004): Collaborating around vertical and horizontal large interactive displays: which way is best?. In: Interacting with Computers 16. pp. 1133-1152.

Ropohl, Günter (1999): Allgemeine Technologie - Eine Systemtheorie der Technik (2. Auflage). München Wien: Carl Hanser Verlag.

Rosson, Mary Beth; Carroll, John (2002): Scenario-Based Design. In: Jacko, Julie A. ; Sears, Andrew (Eds.): The Human-Computer Interaction Handbook. Mahwah, New Jersey: LEA. pp. 1032 - 1050.

Rumbaugh, James; Jacobson, Ivar; Booch, Grady (2005): The Unified Modelling Language Reference Manual, Second Edition. Addison-Wesley.

Sader, Manfred (2001): Manfred Sader: Psychologie der Gruppe, 8. Aufl.. Weinheim: Juventaverlag.

Scheer, August-Wilhelm (1991): Grundlagen der Unternehmensmodellierung. Berlin et al: Springer.

Schmidt, Kjeld (1999): Of maps and scripts - the status of formal constructs in cooperative work. In: Information and Software Technology 41. pp. 319 - 329.

Schmidt, Kjeld; Bannon, Liam (1992): Taking CSCW Seriously. Supporting Articulation Work. In: Computer Supported Cooperative Work (CSCW) 1. pp. 7 - 40.

Schmidt, Kjeld; Simone, Carla (1996): Coordination Mechanisms: Towards a Conceptual Foundation of CSCW Systems Design. In: Computer Supported Cooperative Work: The Journal of Collaborative Computing 5. pp. 155 - 200.

Schmitz-Becker, Harald (2002): Flexibilisierung des Darstellungsumfangs von Diagrammen am Beispiel des Modellierungs-Editors SeeMe. Master Thesis, Universität Dortmund.

Schuman, Sandy (Ed.) (2005): The IAF Handbook of Group Facilitation. San Franscisco: Jossey-Bass.

Schwabe, Gerhard; Streitz, Norbert; Unland, Rainer (Eds.) (2001): CSCW-Kompendium, Lehr- und Handbuch zum computerunterstützten kooperativen Arbeiten. Berlin, Heidelberg: Springer.

Senge, Peter (1990): The fifth discipline. London: Random House.

Sharp, Alec; Mc Dermott, Patrick (2001): Workflow Modeling. Tools for process improvement and application development. Boston: Artech House.

Simonsen, Jesper; Kensing, Finn (1997): Using Ethnography in Contextual Design. In: Communications of the ACM Vol. 40, No.7. pp. 82 - 88.

Sinz, Elmar J. (2002): Konstruktion von Informationssystemen. In: Rechenberg, Peter; Pomberger, Gustav (Eds.): Informatik-Handbuch. München, Wien: Hanser. pp. 1067 - 1084.

Sommerville, Ian (2004): Software Engineering (Seventh Edition). Harlow, England: Addison-Wesley.

Star, Susan Leigh (1989): The Structure of Ill-Structured Solutions: Boundary Objects and Heterogeneous Distributed Problem Solving. In: Huhns, M.; Gasser, L. (Eds.): Distributed Artificial Intelligence 2. Menlo Park, CA: Morgan Kaufman. pp. 37 - 55.

Strauss, Anselm; Corbin, Juliet (1998): Basics of Qualitative Research - Techniques and Procedures for Developing Grounded Theory (Second Edition). Thousand Oaks: Sage Publications.

Streitz, Norbert; Röcker, Carsten; Prante, Thorsten; Stenzel, Richard; van Alphen, Daniel (2003): Situated Interaction with Ambient Information: Facilitating Awareness and Communication in Ubiquitous Work Environments. In: Harris, D.; Duffy, V.; Smith, M.; Stephanidis, C. (Eds.): Human Centred Computing: Cognitive, Social and Ergonomic Aspects. Mahwah, New Jersey: Lawrence Erlbaum Publishers. pp. 133 - 137.

Suchman, Lucy (1987): Plans and situated actions: The problem of human-machine communication. Cambridge, U.K.: Cambridge University Press.

Suchman, Lucy (1994): Do Categories Have Politics? The language/action perspective reconsidered. In: Computer Supported Cooperative Work (CSCW) 2. pp. 177 - 190.

Suchman, Lucy (1995): Making Work Visible. In: Communications of the ACM Volume 38, No. 9. pp. 56 - 64.

Sydow, Jörg (1985): Der soziotechnische Ansatz der Arbeits- und Organisationsgestaltung. Frankfurt, New York: Campus Verlag.

Teufel, Stephanie; Sauter, Christian; Mühlherr, Thomas; Bauknecht, Kurt (1995): Computerunterstützung für die Gruppenarbeit. Bonn: Addison-Wesley.

Trist, Eric (1993): Introduction to Volume II. In: Trist, Eric; Murray, Hugh (Eds.): The social engagement of social science - A Tavistock Anthology. Philadelphia: University of Pennsylvania Press. pp. 36 - 60.

Trist, Eric; Bamforth, Ken (1951): Some social and psychological consequences of the long wall method of coal getting. In: Human Relations 4. pp. 3 - 38.

Trist, Eric; Higgin, G. W.; Murray, Hugh; Pollock, A. B. (1963): Organizational Choice - Capabilities of Groups at the Coal Face under changing Technologies - The Loss, Re-discovery & Transformation of a Work Tradition. London: Tavistock Publications.

Turnwald, Marc (2002): Joint Editing für die kooperative Entwicklung und Vervollständigung semi-strukturierter Modelle. Master Thesis, Universität Dortmund.

Udris, Ivars; Ulich, Eberhard (1987): Organisations- und Technikgestaltung: Prozeß- und partizipationsorientierte Arbeitsanalysen. In: Sonntag, Karl-Heinz (Ed.): Arbeitsanalyse und Technikentwicklung. Beiträge über Einsatzmöglichkeiten arbeitsanalytischer Verfahren bei technisch-organisatorischen Änderungen. Köln: Wissenschaftsverlag Bachem. pp. 49 - 68.

Varela, Francisco (1981): Autonomy and Autopoiesis. In: Roth, Gerhard; Schwegler, Helmut (Eds.): Self-organizing Systems: an interdisciplinary approach. Frankfurt: Campus Verlag. pp. 14 - 23.

Wacker, Stephan (2000): Wahrnehmungspsychologisch fundierte Entwicklung und Umsetzung eines Styleguides für Diagramme soziotechnischer Systeme. Master Thesis, Universität Dortmund.

Watzlawik, Paul; Beavin Bavelas, Janet; Jackson, Don D. (1967): Pragmatics of Human Communication, A study of interactional patterns, pathologies and paradoxes. New York, London: W.W. Norton & Company.

Willke, Helmut (1987): Strategien der Intervention in autonome Systeme. In: Baecker, Dirk; Markowitz, Jürgen; Stichweh, Rudolf; Tyrell, Hartmann; Willke, Helmut (Eds.): Theorie als Passion - Niklas Luhmann zum 60. Geburtstag. Frankfurt am Main: Suhrkamp. pp. 333 - 361.

Willke, Helmut (1994): Systemtheorie II: Interventionstheorie. Stuttgart, Jena: Gustav Fischer Verlag.

Wimmer, Rudolf (1992): Was kann Beratung leisten? In: Wimmer, Rudolf (Ed.): Organisationsberatung - Neue Wege und Konzepte. Wiesbaden: Gabler. pp. 59 - 111.

Winograd, Terry (1988): A Language/Action Perspective on the Design of Cooperative Work. In: Greif, Irene (Ed.): Computer-Supported Cooperative Work: A Book of Readings. San Mateo, California: Morgan Kaufman Publishers. pp. 623 - 653.

Winograd, Terry; Flores, Fernando (1986): Understanding Computers and Cognition: A New Foundation for Design. Norwood, New Jersey: Ablex Publishing Corporation.

Wollnik, Michael (1998): Interventionschancen bei autopoietischen Systemen. In: Götz, Klaus (Ed.): Theoretische Zumutungen. Vom Nutzen der systemischen Theorie für die Managementpraxis. Heidelberg: Carl-Auer-Systeme Verlag. pp. 118 - 159.

Wooldridge, Michael (2002): An Introduction to Multiagent Systems. Chichester, England: John Wiley & Sons, Ltd..

Wulf, Volker (1999): Evolving Cooperation when Introducing Groupware: A Self-Organization Perspective. In: Cybernetics & Human knowing Volume 6, No. 2. pp. 55 - 75.

Wulf, Volker; Jarke, Matthias (2004): The Economics of End-User Development. In: Communications of the ACM Volume 7, No. 9. pp. 41 - 42.

10.2 Electronic Documents and Webpages

HL&D (2002): The Human Leadership and Development Division of the American Society for Quality, The International Association of Facilitators, The Association for Quality and Participation (Eds.) Basic Facilitation Skills. http://www.iaf-world.org/files/public/FacilitatorMnl.pdf (checked: Nov. 2005)

Hoffmann, Marcel (2004): Awareness und Adoption kooperativer Wissensmedien im Kontext informeller Zusammenarbeit. Dissertation, Universität Dortmund. http://hdl.handle.net/2003/19663 (checked: November 2005)

Kruchten, Philippe (2003): What is the Rational Unified Process? http://www-128.ibm.com/developerworks/rational/library/content/RationalEdge/feb03/WhatisRUP_TheRationalEdge_Feb2003.pdf (checked: November 2005)

Loser, Kai-Uwe (2005): Unterstützung der Adoption kommerzieller Standardsoftware durch Diagramme. Dissertation, Universität Dortmund. http://hdl.handle.net/2003/21659 (checked: Novermber 2005)

Webpage ACM SigSoft; http://www.sigsoft.org/about/index.htm (checked: November 2005)

Webpage Agile Manifesto; http://www.agilemanifesto.org/principles.html (checked: November 2005)

Webpage CPSR; http://cpsr.org/program/workplace/PD.html (checked: April 2004)

Webpage CSCW-Workshop 2004; http://www.edgelab.ca/CSCW/Workshop2004/ (checked Nov. 2005)

Webpage GI; http://www.gi-ev.de/service/informatiklexikon/informatiklexikon-detailansicht/meldung/27/ (checked: November 2005)

Webpage IAF; http://www.iaf-world.org/i4a/pages/index.cfm?pageid=1 (checked: August 2005)

Webpage Schwabe; http://www.schwabe.de/content/presse/pressefotos_arzneipflanzen.php?navtext=Pressefotos (checked: August 2005)

Webpage SeeMe-Editor; http://www.imtm-iaw.rub.de/projekte/seeme/index.html (checked: Nov. 2005)

Webpage SUN, Java Web Start; http://java.sun.com/developer/technicalArticles/WebServices/JWS_2/JWS_White_Paper.pdf (checked: Oktober 2005)

Webpage Weimar; http://www.planet-weimar.de/english/englisch.html (checked: August 2005)

10.3 Author Index

10.4 List of Abbreviations

Acronym	Meaning
ANT	Actor Network Theory
API	Application Programming Interface
AR	Action Research
ARIS	Architecture of Integrated Information Systems (originally in German: Architektur integrierter Informationssysteme)
AST	Adaptive Structuration Theory
BMBF	German Ministry for Education and Research (Bundesministerium für Bildung und Forschung)
BPM	Business Process Modelling
BPR	Business Process Reengineering

Acronym	Meaning
BSCW	Basic Support for Cooperative Work
CD	Contextual Design
CI	Contextual Inquiry
CIO	Chief Information Officer
CSCL	Computer Supported Collaborative Learning
CSCW	Computer Supported Cooperative Work
DTD	Data Type Definition
ELISE	Elektronisches Inhaltsverzeichnis System (electronic table of contentssystem)
eEPC	Extended Event-Process-Chain
EPC	Event-Process-Chain
ETHICS	Effective Technical and Human Design of Computer-based Systems
GIF	Graphics Interchange Format
GSM	Global System for Mobile Communications
GPRS	General Packet Radio Service
HCI	Human-Computer-Interaction
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
ISD	Information System Design
IT	Information Technology
JAD	Joint Application Design
JNLP	JAVA™ Network Launching Protocol
JPG	Joint Pictures Group
JPEG	Joint Pictures Expert Group
KMS	Knowledge Management System
LAN	Local Area Network
KBS	Knowledge Based System
MMI	Man-Machine-Interaktion
MUST	Danish acronym for „Theories of and methods for design activities”
PD	Participatory Design
PNG	Portable Network Graphics
QWL	Quality of Working Life
RE	Requirements Engineering, Requirements Elicitation
RUP	Rational Unified Process
SOAP	Simple Object Access Protocol
SSM	Soft Systems Methodology
STEPS	Softwaretechnik für Evolutionäre Partizipative Systemgestaltung
STS	Socio-technical System
StSd	Socio-technical Self-description
STWT	Socio-technical Walkthrough
SWE	Software Engineering
TOC	Table of Content
UB	Universitätsbibliothek (German for University Library)
UML	Unified Modelling Language
UMTS	Universal Mobile Telecommunications System
XML	Extended Meta Language
XP	Extreme Programming
ZID	ToC-Service (German: Zeitschrifteninformationsdienst)

10.5 List of Terms Defined for this Thesis

10.6 List of Examples from Case Studies

10.7 List of Tables[34]

[34] Table 33 - Table 40 are included in a supplementary appendix in chapter 11

10.8 List of Figures[35]

[35] Figure 61 - Figure 110 are included in the supplementary appendix in chapter 11

11 Supplementary Appendices

11.1 Appendix – Overview of Empirical Work and Material

This appendix contains overviews of the empirical material collected during the two case studies described in sections 6.2 and 6.3.

11.1.1 Analysis Phase in Case Study 1

Interview with Experts, 23 July 2002

Participants: 2 researchers, project-leader „Westkreis", 1 manager from central office;

Goal / Topics: Gather initial information about structures and processes of „Westkreis"

Material available for analysis: document that contains information about „Westkreis" presented according to KABA guideline [Dunckel (2003)]; organization chart of „Westkreis".

Observational Interviews, 25 July 2002, 6 – 20 Sep 2002

Participants: 2 researchers conducted a total of 8 observational interviews with 6 dispatchers and 2 drivers.

Goal / Topics: Gather information about structure and processes of „Westkreis".

Material available for analysis: Data from 8 working days was collected and documented in form of KABA-sheets [Dunckel (2003)]; photos from various situations and documents; diagram representing the status quo as seen by the researchers.

11.1.2 Workshops Conducted in Case Study 1

STWT-1, 17 Dec 2002

Participants: 3 dispatchers, 2 drivers, 1 project-leader „Westkreis", 1 software engineer, 2 researchers, 1 assistant to handle the editor and draw diagrams

Goal / Topics: Presentation, discussion and correction of findings from analysis phase; gathering of input to a list of information-flows for the project.

Material available for analysis: Preparatory documents detailing the context, goals and structure of the workshop; diagram that served as input into the workshop; audio-recording of the workshop; diagrams created during the workshop; document containing immediate feedback of researchers after the workshop.

STWT-2, 05 Feb 2003

Participants: 3 dispatchers, 2 drivers, 1 project-leader „Westkreis", manager from the main office, 1 software engineer, 2 researchers, assistant to operate the editor and draw diagrams

Goal / Topics: Common understanding of the way in which SpiW-Com can support the cooperative work of drivers and dispatchers.

Material available for analysis: Preparatory documents detailing the context, goals and structure of the workshop; Audio-recording of the workshop; diagrams created during the workshop; document containing immediate feedback of researchers after the workshop.

Meeting on 28 May 2003

Participants: 2 managers from the main office, 2 researcher

Goal / Topics: Evaluation of the GUI screenshots

Material available for analysis: Screenshots, Diagrams created in preparation of and during the meeting.

Meeting on 6 June 2003

Participants: 1 manager from the main office, 1 software engineer, 2 researchers

Goal / Topics: Evaluation of GUI Screenshots (continued)

Material available for analysis: Screenshots, Diagrams created in preparation of and during the meeting.

STWT-3, 18 July 2003 and 29 July 2003 (cont)

Participants: 1 dispatcher (on July 18th only), 1 project-leader „Westkreis", 1 manager from the main office, 1 software engineer, 2 researchers

Goal / Topics: Discussion of GUI-Screenshots in context of work-procedures

Material available for analysis: Handout for participants containing Diagrams, screenshots and topic of workshop; textual protocol; diagrams modified during the workshop.

Several Meetings in February and March 2004

Participants: 1 – 2 managers from the main office, 1 – 2 software engineers, 2 researchers

Goal / Topics: Preparation of STWT-4; updating diagrams; testing software; defining scenarios.

Material available for analysis: Passages in research diary.

Meeting on 5 March 2004

Participants: Project leader „Westkreis"; 2 researchers

Goal / Topics: Detailing new work-procedures for dispatchers

Material available for analysis: Preparatory documents detailing goals and structure of the session; Diagrams created / modified during the meeting; Feedback from dispatcher concerning diagrams.

STWT-4, 19 March 2004

Participants: 1 project-leader „Westkreis", 2 drivers, 1 manager and 1 staff member from the main office, 2 software engineers, 2 researchers

Goal / Topic: Training of drivers in using the mobile client of SpiW-Com; discussion about open issues concerning the usage of SpiW-Com.

Material available for analysis: Preparatory documents detailing goals, structure and methods for the workshop and including scenarios to be used in the workshop; diagrams created / modified during the workshop; photos from all flipcharts / boards written during the workshop; video recording; transcripts of interviews conducted with the participants a few days after the workshop.

11.1.3 Workshops Conducted in Case Study 2

STWT-1, 15 July 2003

Participants: Head of team, 2 student workers from library team, 7 scientific team members (one of them acting as facilitator)

Goal / Topics: Kick-off meeting for the project.

Material available for analysis: Diagrams containing the status quo of work procedures in the library, which were prepared by library team as input; handout with overview of diagrams; modified diagrams documenting the result of the discussion; minutes summarizing the results; video recording of workshop; document containing feedback of participants taken a day after the workshop.

STWT-2, 11 Sep 2003

Participants: Head of team, 1 student workers from library team, 6 scientific team members (one of them acting as facilitator); software engineer.

Goal / Topics: Collection of further requirements for the socio-technical setting; discussion of team leader's ideas for working procedures.

Material available for analysis: Diagram prepared by head of team serving as input; poster summarizing results of the previous workshop; diagrams created during the workshop; minutes summarizing the results; video recording of workshop.

STWT-3, 27 Nov 2003

Participants: Head of team, 1 student workers from library team, 9 scientific team members (one of them acting as facilitator), software engineer.

Goal / Topics: 1. Description of the „lifecycle" of a bibliographical reference within the team's work procedures; collection of technical options to support the activities occurring during the lifecycle.

Material available for analysis: Poster summarizing the results of the previous workshop; diagram created during the workshop describing the envisioned technically supported procedures; photos of all flipcharts and boards used and created during the workshop; notes taken by researcher during the workshop.

5 Technical meetings, December 2003 – January 2004

Participants: In each of the 5 meetings one team member with special expertise for a certain technical solution met with the software engineer; in some of the meetings an additional team member was present to create the diagrams during the discussion.

Goal / Topics: Outline computer supported work-procedures for each of the five technical options.

Material available for analysis: Diagram showing the activities to be supported serving as input in each of the 5 meetings; 5 resulting diagrams – one for each technical option – showing the computer supported work procedures.

STWT-4, 30 Jan 2004

Participants: Head of team; 8 members of scientific team (one of them acting as a facilitator); software engineer.

Goal / Topics: Selection of the technical option (one out of five) that supports the intended work-procedures best.

Material available for analysis: Diagrams resulting from the 5 technical meetings and serving as input to the workshop; preparatory documents including goal and structure of the workshop; photos of setting; photos of all flipcharts and boards used and created during the workshop; video-recording including feedback of participants on usefulness of diagrams;

Workshop for library team, 4 April 2004

Participants: 2 student workers of library team; software engineer.

Goal / Topics: To inform the library team about the project; to outline the effects of the new technical system on their work-procedures.

Material available for analysis: Diagrams that were used as input to workshop; diagram created during the workshop; audio-recording of workshop.

Workshop for Project Management, 30 March 2004

Participants: Head of team, 1 member of scientific team; software engineer.

Goal / Topics: To agree on a version for the computer supported work-procedures.

Material available for analysis: Diagrams resulting from previous (partly informal) discussions; diagrams modified according to the discussions in the workshop.

Technical meeting Filemaker®, 1 April 2004

Participants: 2 members of scientific team; software engineer.

Goal / Topics: Discussion of technical prototype based on Filemaker®

Material available for analysis: Diagrams showing technically supported work-procedures and prototype serving as input to the meeting; entity-relationship diagram for database design; comments on prototype and modified diagrams resulting from the discussion.

Technical Meeting K2, 8 June 2004

Participants: 3 members of scientific team; software engineer.

Goals / Topics: Find a technical solution for data exchange between Filemaker® and K2.

Material available for Analysis: Diagram with computer supported work procedures and FM-based prototype serving as input; notes in research diary.

Qualification Workshop, 30 June 2004

Participants: Head of team; 1 student worker from library team; 5 members of scientific team (one of them acting as the facilitator).

Goals / Topics: Participants should learn how to handle the ToC-service ELISE; participants should agree on necessary organizational rules.

Material available for analysis: Prototype ELISE; diagrams illustrating the corresponding work-procedures serving as an input; video recording of workshop; photos of all flipcharts and boards created and used during the workshop; diagrams created during the workshop.

Re-Kick-Off Meeting, 4 April 2005

Participants: 2 members of scientific staff, researcher

Goal / Topics: „Kick-off" of relaunch after break in project

Material available for analysis: Diagrams taken from user manual that were used for the discussion ([Ho (2004), p. 118, 119]; questionnaire filled in by the 2 participants; notes in research diary.

Qualification Workshop, 12 April 2005

Participants: 2 members of scientific staff, one student worker from library, researcher

Goals / Topics: Explanation of ELISE to student worker

Material available for analysis: Questionnaire filled in by 2 members of scientific staff and by student worker; notes in research diary.

Discussion of status quo with head of team, 19 April 2005

Participants: Head of team, 2 members of scientific staff, researcher

Goals / Topics: presentation and discussion of status quo concerning the re-launch of the project; decision about going operational

Material available for Analysis: Questionnaire filled in by 1 member of scientific team and head of team; notes in research diary.

(Re-) Qualification of Scientific Staff, 26 April 2005

Participants: Head of team, 6 members of scientific staff, student worker from library, researcher

Goals / Topics: Presentation of CSCW-system ELISE and associated organizational procedures as they were agreed upon in June 2004 (see above). Some of the scientific staff had not participated in the workshop in June 2004, and the others had forgotten much of what had been decided. So the purpose of the workshop was to (re-) qualify the members of the scientific staff in using ELISE. At the end of this session, the decision was taken to start using ELISE in May and to stop the circulation of paper files at the same time.

Material available for Analysis: Questionnaire filled in by 5 members of scientific team and by head of team; notes in research diary.

Discussion during team meeting, 25 May 2005

Participants: 5 members of scientific staff, researcher.

Goals / Topics: Discussion of experiences regarding use of ELISE.

Material available for Analysis: Diagrams used for discussion; notes in research diary.

Presentation of new Version of ELISE, 29 June 2005

Participants: Head of team; 4 members of scientific staff, researcher.

Goals / Topics: Based on the feedback of the users, a major redesign of the GUI was done; the workshop had the goal to present the new GUI and put the functions into relation with the agreed work procedures.

Material available for Analysis: Diagrams; notes in research diary.

Evaluation workshop, 22 August 2005

Participants: head of team, 5 members of scientific team, student worker from library, researcher

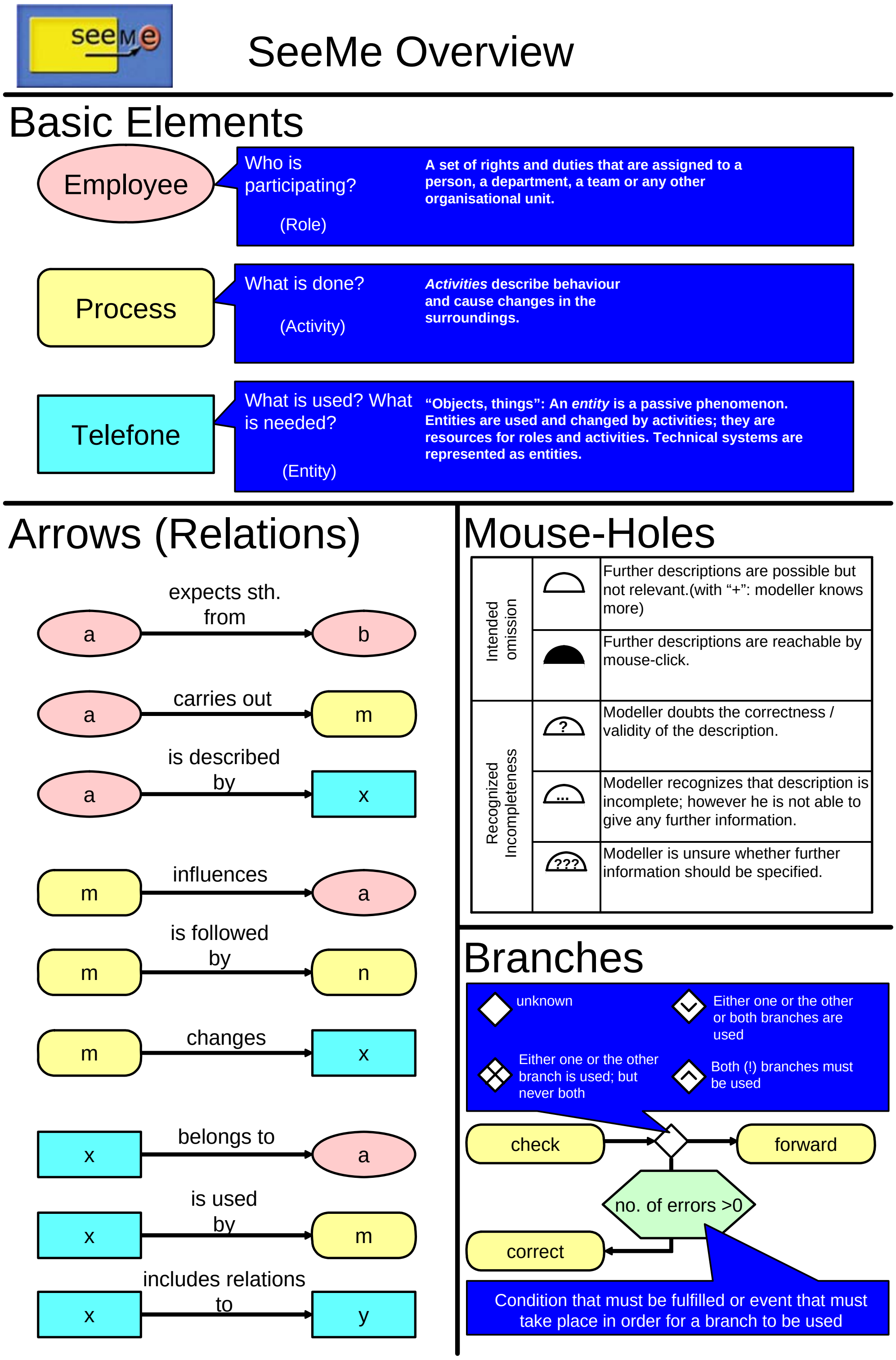
Goals / Topics: Collection of feedback regarding CSCW-system, its actual use and its integration into work procedures. This workshop marks the end of the case study as far as it is relevant for this thesis.

Material available for Analysis: Audio recording of workshop; diagrams; notes in research diary.

11.2 Appendix – Overview of SeeMe Notation

The modeling notation SeeMe was used for the semi-structured diagrams in the case studies. Section 4.3.5 (pp. 126) contains a brief introduction including references to literature that explains the concepts of the modeling language and its theoretical background. Section 5.2.2 (pp. 137) discusses modeling constructs that are especially relevant for CSCW-projects.

Figure 61 provides an overview of the notation; the level of detail equals that of the introduction given to the participants of the workshops in case study 1 „Mobile Communication System". More information can be found on the web page [Webpage SeeMe-Editor].



SeeMe Overview

Basic Elements

Employee

Who is participating?

(Role)

A set of rights and duties that are assigned to a person, a department, a team or any other organisational unit.

Process

What is done?

(Activity)

Activities describe behaviour and cause changes in the surroundings.

Telefone

What is used? What is needed?

(Entity)

"Objects, things": An *entity* is a passive phenomenon. Entities are used and changed by activities; they are resources for roles and activities. Technical systems are represented as entities.

Arrows (Relations)

Mouse-Holes

Intended omission	◠	Further descriptions are possible but not relevant.(with "+": modeller knows more)
	◗	Further descriptions are reachable by mouse-click.
Recognized Incompleteness	?	Modeller doubts the correctness / validity of the description.
	...	Modeller recognizes that description is incomplete; however he is not able to give any further information.
	???	Modeller is unsure whether further information should be specified.

Branches

◇ unknown

Either one or the other or both branches are used

Either one or the other branch is used; but never both

Both (!) branches must be used

Condition that must be fulfilled or event that must take place in order for a branch to be used

Figure 61: SeeMe - Overview of Basic Elements

11.3 Appendix – Diagrams from the Case Studies

This appendix contains a collection of diagrams created during the case Studies. Since it would exceed the capacity of this document by far to print all diagrams created and discussed during the projects, a subset of diagrams was selected according to the following criteria.

a) Skimming through the diagrams should give an impression of their evolvement in the course of the project.

b) The diagrams should support the reading of chapters 7 and 8 without interrupting the text. To illustrate an argument in the text of the chapters, translated details of the diagrams are included where needed.

11.3.1 „Mobile Communication System”

Figure	Content / Purpose
Figure 62: Result of Ethnographically Oriented Analysis	
	The Diagram was drawn by the researchers to present the result of the analysis of the status quo to the participants of STWT 1
Figure 64: Result of STWT-1	
	The diagram shows the result of STWT 1. The information that should be exchanged between drivers and dispatchers has been detailed.
Figure 66 – Figure 70	
	These diagrams result from STWT-2. They contain early ideas of technically supported work-procedures as well as requirements for SpiW-Com.
Figure 71 – Figure 73	
	Three examples of numerous diagrams that were created in parallel with the design of the software. The diagrams contain descriptions of data and functions provided by SpiW-Com.
Figure 74 – Figure 80	
	These diagrams were created and used during the validation phase of the GUI-prototype. The entities „Pocket PC” in the diagrams contain hyperlinks to the corresponding masks of the GUI. Most of the comments included in the diagrams result from STWT-3 in which they were discussed.
Figure 81 – Figure 85	
	These diagrams document the result of the fourth STWT which was a workshop for qualification of the participants. The diagrams contain agreements about the usage of the mobile communication system SpiW-Com.

Table 33: Overview of Diagrams in ‘Mobile Communication System’

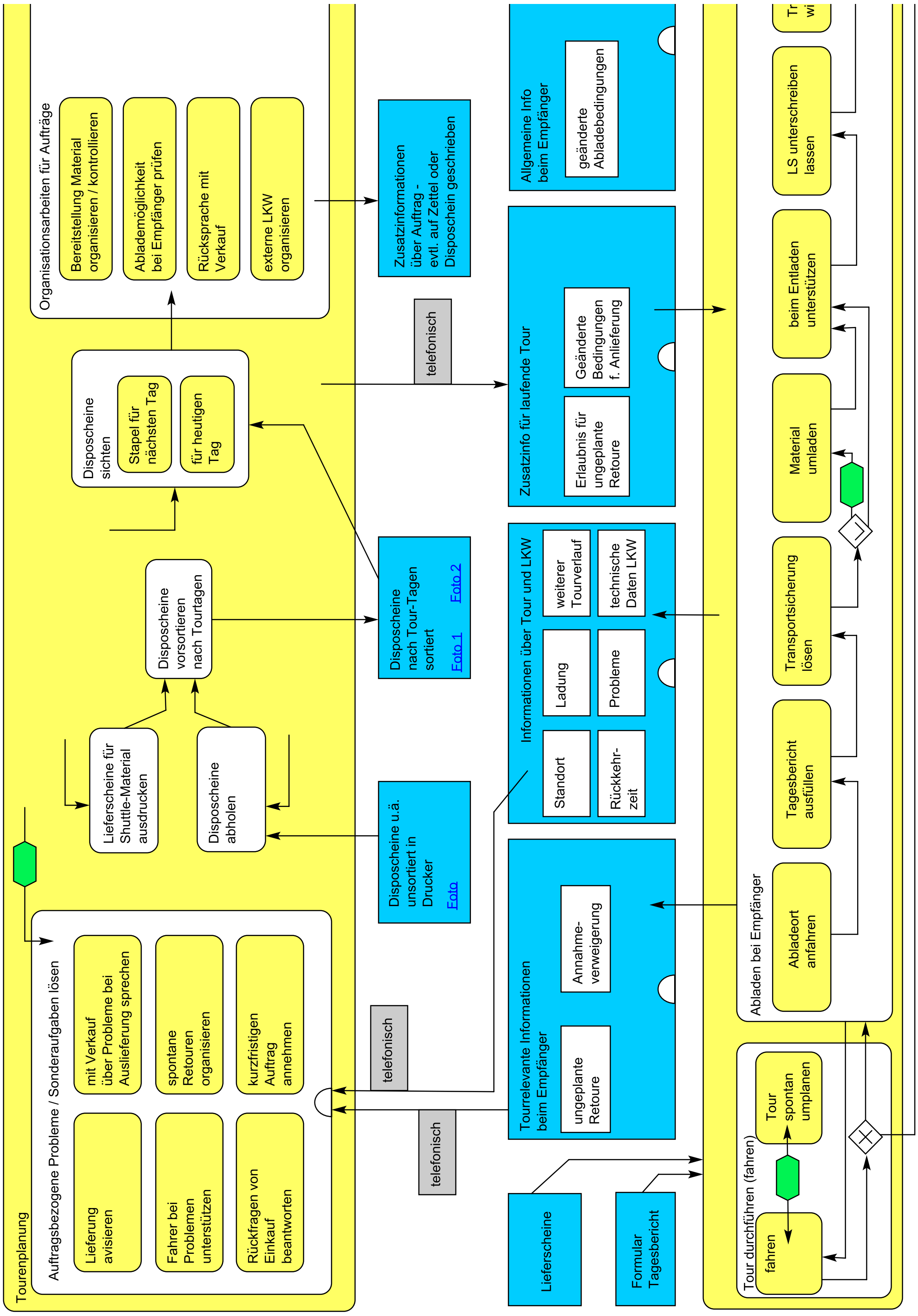

Tourenplanung
Auftragsbezogene Probleme / Sonderaufgaben lösen
Lieferung avisieren
mit Verkauf über Probleme bei Auslieferung sprechen
Fahrer bei Problemen unterstützen
spontane Retouren organisieren
Rückfragen von Einkauf beantworten
kurzfristigen Auftrag annehmen
Lieferscheine für Shuttle-Material ausdrucken
Disposcheine abholen
Disposcheine vorsortieren nach Tourtagen
Disposcheine sichten
Stapel für nächsten Tag
für heutigen Tag
Organisationsarbeiten für Aufträge
Bereitstellung Material organisieren / kontrollieren
Ablademöglichkeit bei Empfänger prüfen
Rücksprache mit Verkauf
externe LKW organisieren
Zusatzinformationen über Auftrag - evtl. auf Zettel oder Disposchein geschrieben
telefonisch
Disposcheine nach Tour-Tagen sortiert
Foto 1
Foto 2
Disposcheine u.ä. unsortiert in Drucker
Foto
telefonisch
telefonisch
Allgemeine Info beim Empfänger
geänderte Abladebedingungen
Zusatzinfo für laufende Tour
Erlaubnis für ungeplante Retoure
Geänderte Bedingungen f. Anlieferung
Informationen über Tour und LKW
Standort
Ladung
weiterer Tourverlauf
Rückkehr-zeit
Probleme
technische Daten LKW
Tourrelevante Informationen beim Empfänger
ungeplante Retoure
Annahme-verweigerung
Lieferscheine
Formular Tagesbericht
Tour durchführen (fahren)
fahren
Tour spontan umplanen
Abladen bei Empfänger
Abladeort anfahren
Tagesbericht ausfüllen
Transportsicherung lösen
Material umladen
beim Entladen unterstützen
LS unterschreiben lassen
Tr
wi

Figure 62: Result of Ethnographically Oriented Analysis (Part 1 of 2)

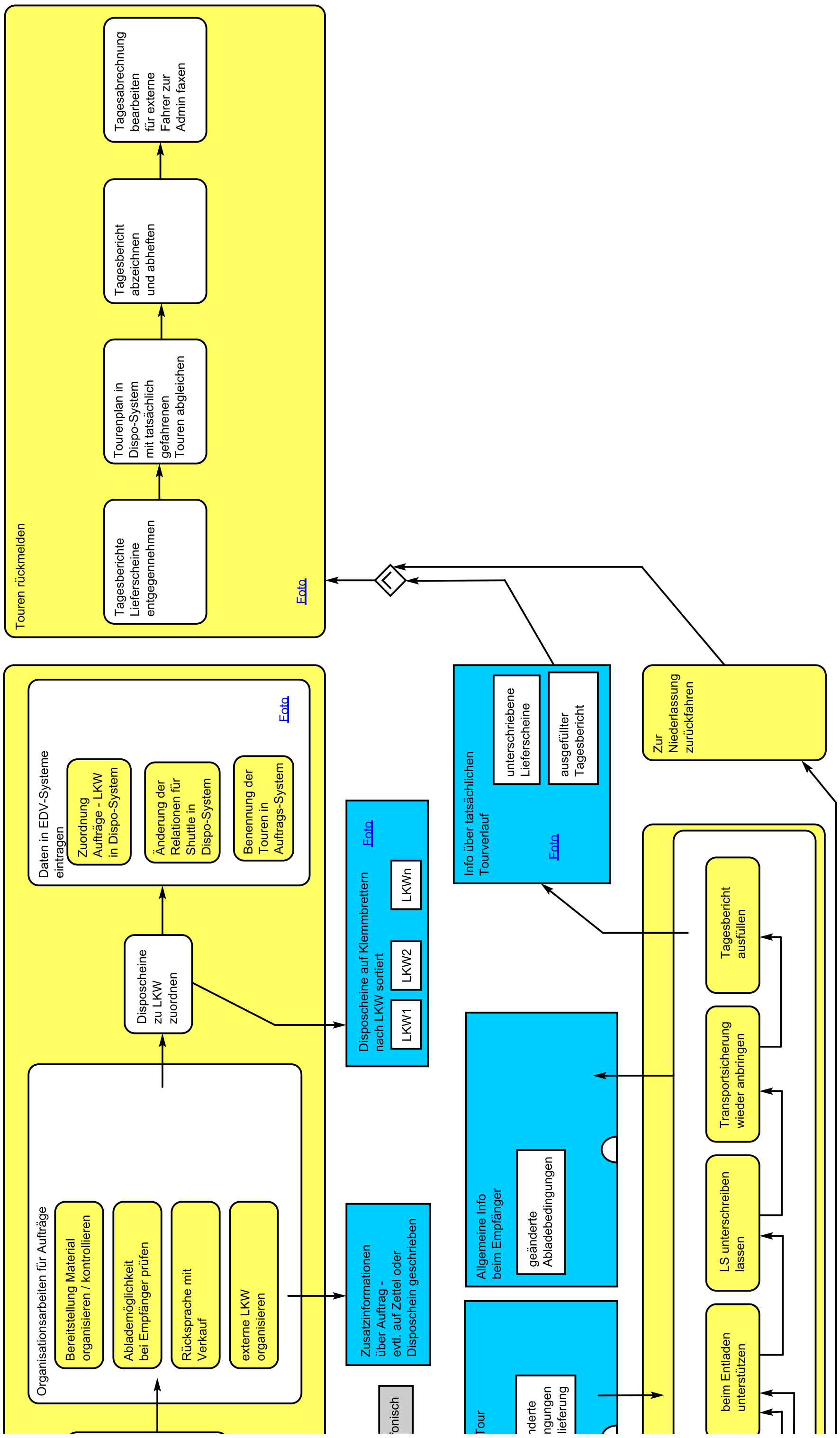

Touren rückmelden
Tagesberichte Lieferscheine entgegennehmen
Tourenplan in Dispo-System mit tatsächlich gefahrenen Touren abgleichen
Tagesbericht abzeichnen und abheften
Tagesabrechnung bearbeiten für externe Fahrer zur Admin faxen
Foto
Organisationsarbeiten für Aufträge
Bereitstellung Material organisieren / kontrollieren
Ablademöglichkeit bei Empfänger prüfen
Rücksprache mit Verkauf
externe LKW organisieren
Disposcheine zu LKW zuordnen
Daten in EDV-Systeme eintragen
Zuordnung Aufträge - LKW in Dispo-System
Änderung der Relationen für Shuttle in Dispo-System
Benennung der Touren in Auftrags-System
Foto
Disposcheine auf Klemmbrettern nach LKW sortiert
LKW1
LKW2
LKWn
Foto
Zusatzinformationen über Auftrag - evtl. auf Zettel oder Disposchein geschrieben
önisch
Info über tatsächlichen Tourverlauf
unterschriebene Lieferscheine
ausgefüllter Tagesbericht
Foto
Allgemeine Info beim Empfänger
geänderte Abladebedingungen
Tour
nderte
ngungen
lieferung
Zur Niederlassung zurückfahren
Tagesbericht ausfüllen
Transportsicherung wieder anbringen
LS unterschreiben lassen
beim Entladen unterstützen

Figure 63: Result of Ethnographically Oriented Analysis (Part 2 of 2)

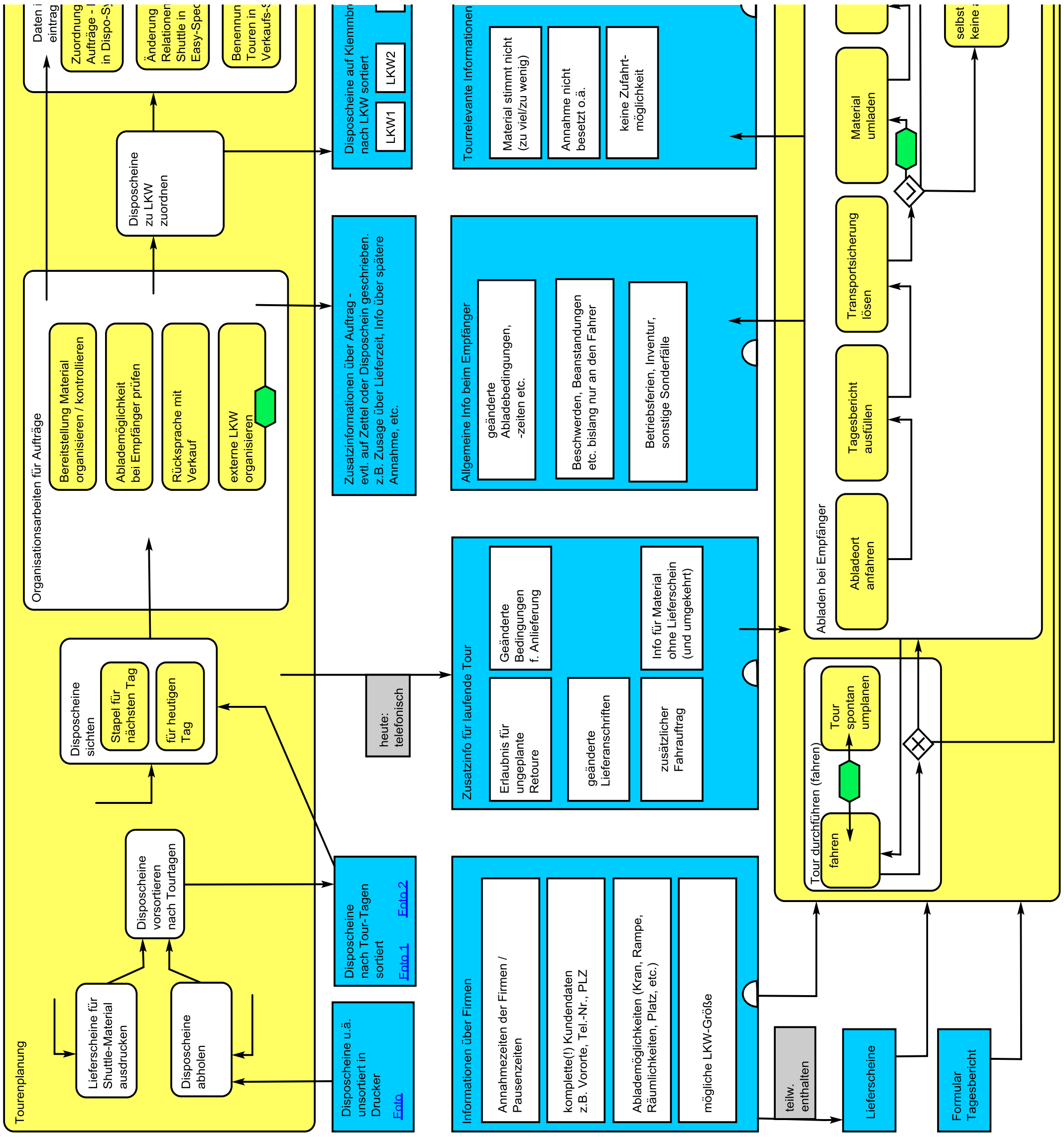

Tourenplanung
Lieferscheine für Shuttle-Material ausdrucken
Disposcheine abholen
Disposcheine vorsortieren nach Tourtagen
Disposcheine sichten
Stapel für nächsten Tag
für heutigen Tag
Organisationsarbeiten für Aufträge
Bereitstellung Material organisieren / kontrollieren
Abklademöglichkeit bei Empfänger prüfen
Rücksprache mit Verkauf
externe LKW organisieren
Disposcheine zu LKW zuordnen
Daten i eintrag
Zuordnung Aufträge - I in Dispo-Sy
Änderung Relationen Shuttle in Easy-Spec
Benennun Touren in Verkaufs-S
Disposcheine u.ä. unsortiert in Drucker
Foto
Disposcheine nach Tour-Tagen sortiert
Foto 1
Foto 2
heute: telefonisch
Zusatzinformationen über Auftrag - evtl. auf Zettel oder Disposchein geschrieben. z.B. Zusage über Lieferzeit, Info über spätere Annahme, etc.
Disposcheine auf Klemmbr nach LKW sortiert
LKW1
LKW2
Informationen über Firmen
Annahmezeiten der Firmen / Pausenzeiten
komplette(!) Kundendaten z.B. Vororte, Tel.-Nr., PLZ
Abladmöglichkeiten (Kran, Rampe, Räumlichkeiten, Platz, etc.)
mögliche LKW-Größe
Zusatzinfo für laufende Tour
Erlaubnis für ungeplante Retoure
Geänderte Bedingungen f. Anlieferung
geänderte Lieferanschriften
zusätzlicher Fahrauftrag
Info für Material ohne Lieferschein (und umgekehrt)
Allgemeine Info beim Empfänger
geänderte Abladebedingungen, -zeiten etc.
Beschwerden, Beanstandungen etc. bislang nur an den Fahrer
Betriebsferien, Inventur, sonstige Sonderfälle
Tourrelevante Informationen
Material stimmt nicht (zu viel/zu wenig)
Annahme nicht besetzt o.ä.
keine Zufahrt-möglichkeit
teilw. enthalten
Lieferscheine
Formular Tagesbericht
Tour durchführen (fahren)
fahren
Tour spontan umplanen
Abladen bei Empfänger
Abladeort anfahren
Tagesbericht ausfüllen
Transportsicherung lösen
Material umladen
selbst keine

Figure 64: Result of STWT-1 (Part 1 of 2)

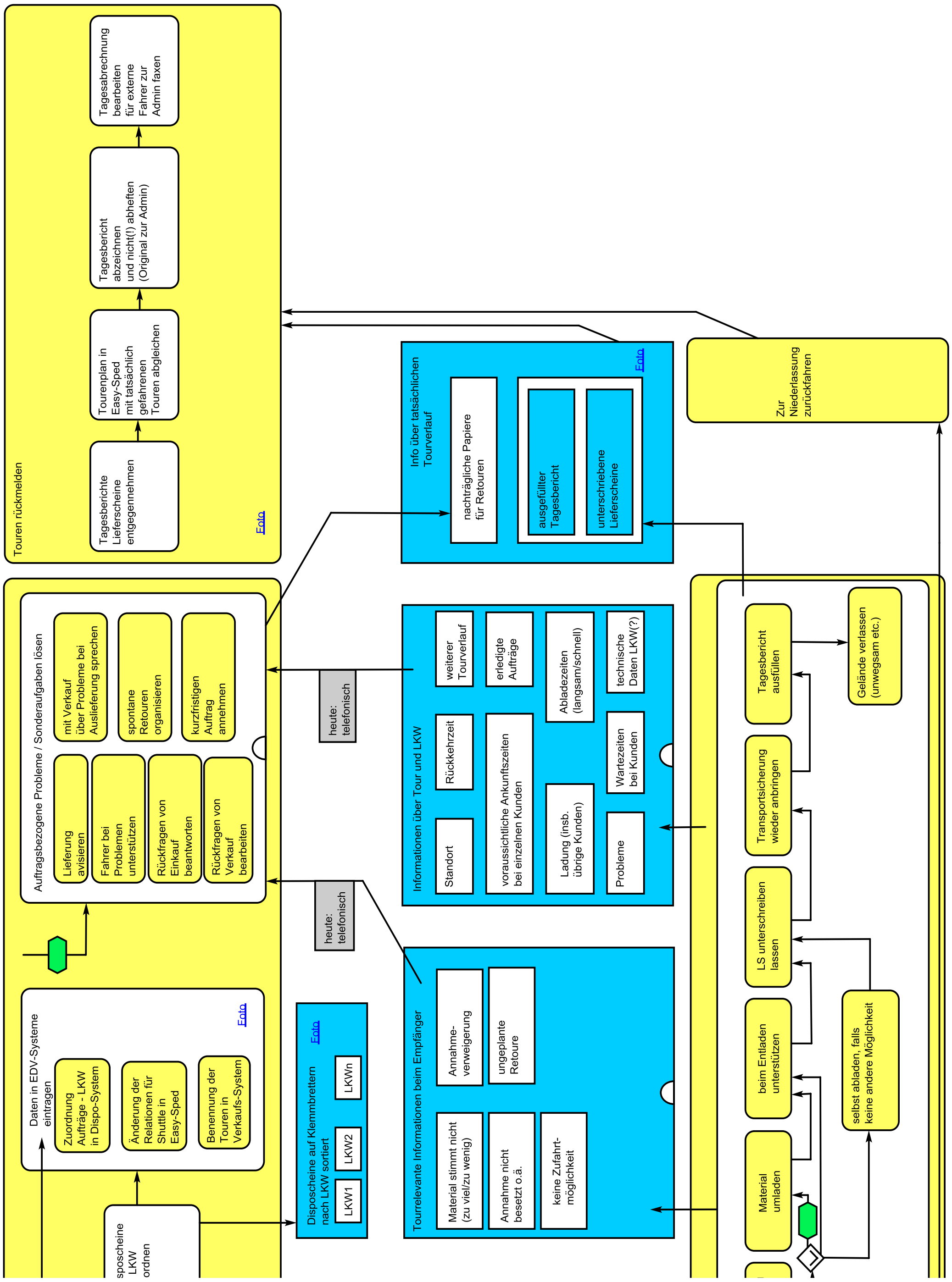

Touren rückmelden
Tagesberichte Lieferscheine entgegennehmen
Tourenplan in Easy-Sped mit tatsächlich gefahrenen Touren abgleichen
Tagesbericht abzeichnen und nicht(!) abheften (Original zur Admin)
Tagesabrechnung bearbeiten für externe Fahrer zur Admin faxen
Foto
Info über tatsächlichen Tourverlauf
nachträgliche Papiere für Retouren
ausgefüllter Tagesbericht
unterschriebene Lieferscheine
Foto
Zur Niederlassung zurückfahren
Auftragsbezogene Probleme / Sonderaufgaben lösen
mit Verkauf über Probleme bei Auslieferung sprechen
spontane Retouren organisieren
kurzfristigen Auftrag annehmen
Lieferung avisieren
Fahrer bei Problemen unterstützen
Rückfragen von Einkauf beantworten
Rückfragen von Verkauf bearbeiten
heute: telefonisch
heute: telefonisch
Informationen über Tour und LKW
Standort
Rückkehrzeit
weiterer Tourverlauf
voraussichtliche Ankunftszeiten bei einzelnen Kunden
erledigte Aufträge
Ladung (insb. übrige Kunden)
Abladezeiten (langsam/schnell)
Probleme
Wartezeiten bei Kunden
technische Daten LKW(?)
Daten in EDV-Systeme eintragen
Zuordnung Aufträge - LKW in Dispo-System
Änderung der Relationen für Shuttle in Easy-Sped
Benennung der Touren in Verkaufs-System
Foto
Disposcheine auf Klemmbrettern nach LKW sortiert
LKW1
LKW2
LKWn
Foto
sposcheine LKW ordnen
Tourrelevante Informationen beim Empfänger
Material stimmt nicht (zu viel/zu wenig)
Annahme nicht besetzt o.ä.
keine Zufahrt-möglichkeit
Annahme-verweigerung
ungeplante Retoure
Tagesbericht ausfüllen
Gelände verlassen (unwegsam etc.)
Transportsicherung wieder anbringen
LS unterschreiben lassen
beim Entladen unterstützen
selbst abladen, falls keine andere Möglichkeit
Material umladen

Figure 65: Result of STWT-1 (Part 2 of 2)

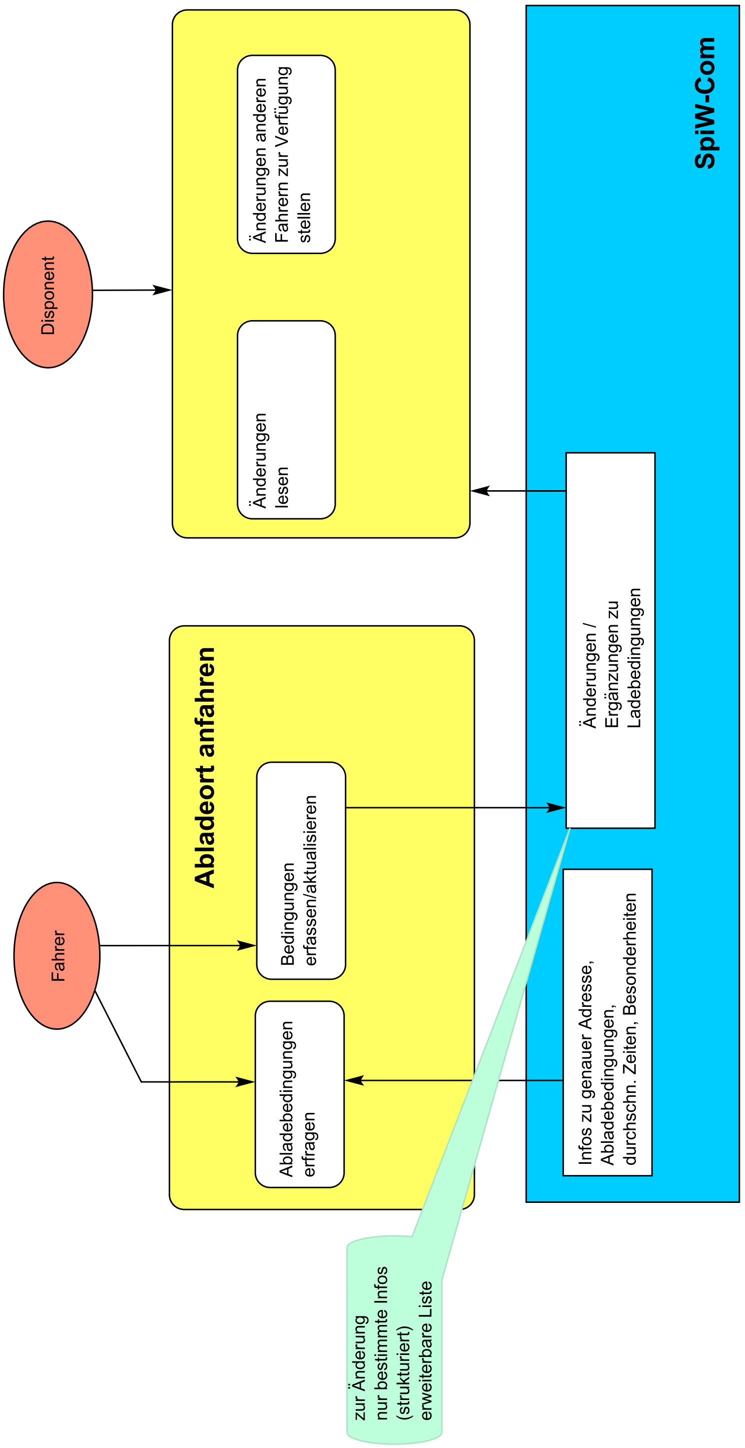

Abladeort anfahren
Fahrer
Disponent
Bedingungen erfassen/aktualisieren
Abladebedingungen erfragen
Änderungen lesen
Änderungen anderen Fahrern zur Verfügung stellen
SpiW-Com
Änderungen / Ergänzungen zu Ladebedingungen
Infos zu genauer Adresse, Abladebedingungen, durchschn. Zeiten, Besonderheiten
zur Änderung
nur bestimmte Infos
(strukturiert)
erweiterbare Liste

Figure 66: Arrival at Customer's Site

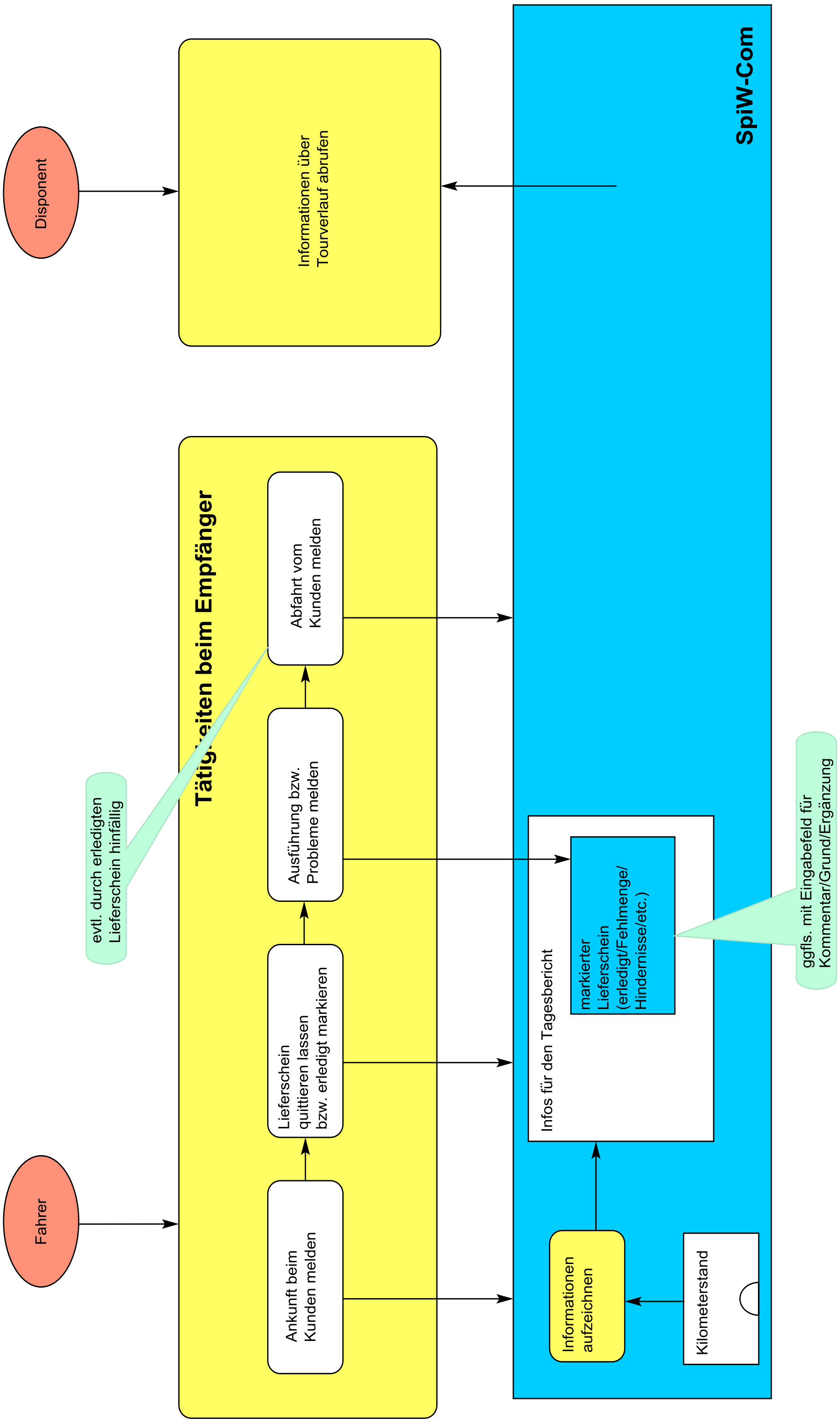
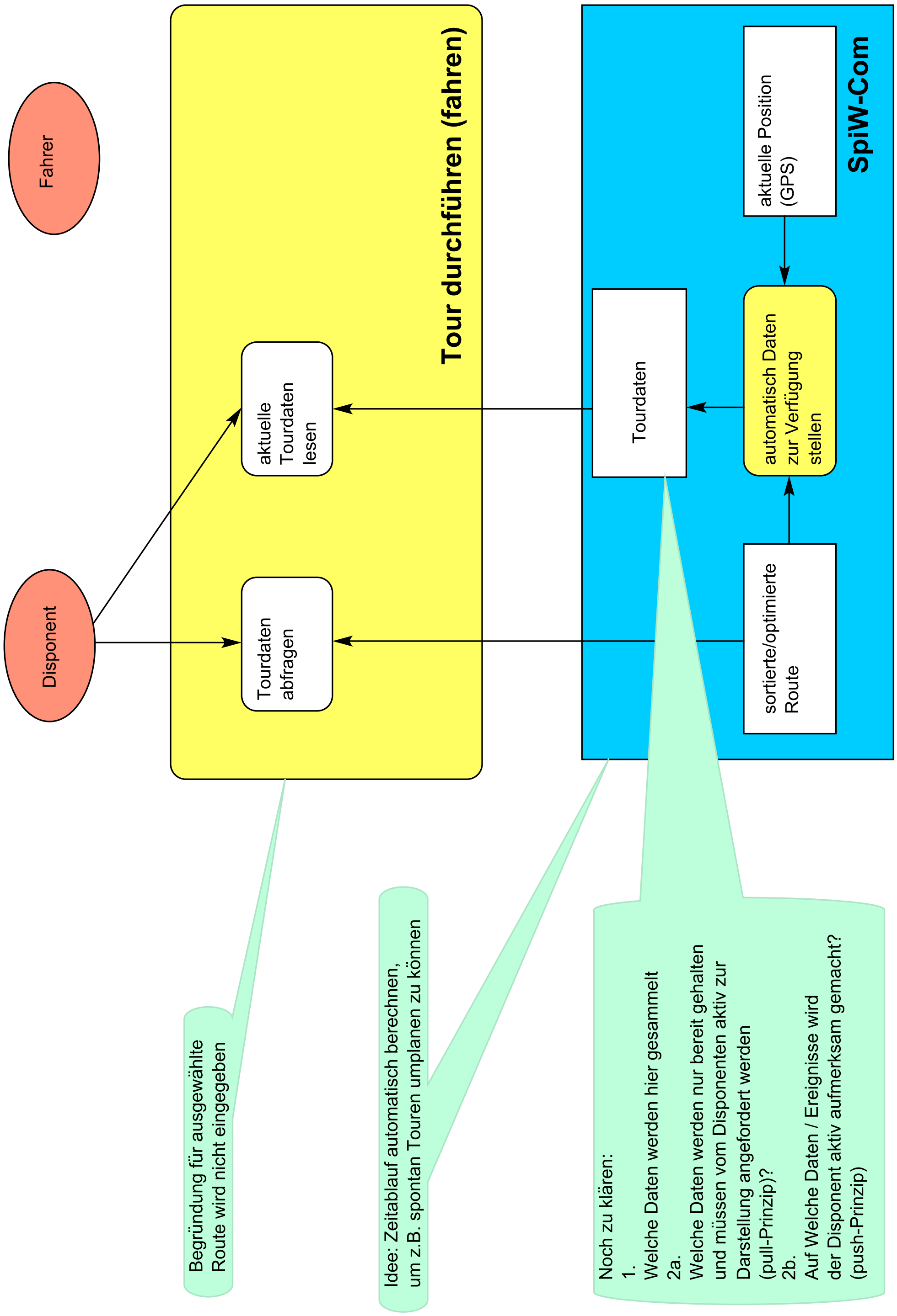

Disponent
Informationen über Tourverlauf abrufen
SpiW-Com
Fahrer
Tätigkeiten beim Empfänger
Ankunft beim Kunden melden
Lieferschein quittieren lassen bzw. erledigt markieren
Ausführung bzw. Probleme melden
Abfahrt vom Kunden melden
evtl. durch erledigten Lieferschein hinfällig
Informationen aufzeichnen
Kilometerstand
Infos für den Tagesbericht
markierter Lieferschein (erledigt/Fehlmenge/ Hindernisse/etc.)
ggfls. mit Eingabefeld für Kommentar/Grund/Ergänzung

Figure 67: Support for Generating the Daily Report



Begründung für ausgewählte Route wird nicht eingegeben

Idee: Zeitablauf automatisch berechnen, um z.B. spontan Touren umplanen zu können

Noch zu klären:
1.
Welche Daten werden hier gesammelt
2a.
Welche Daten werden nur bereit gehalten und müssen vom Disponenten aktiv zur Darstellung angefordert werden (pull-Prinzip)?
2b.
Auf Welche Daten / Ereignisse wird der Disponent aktiv aufmerksam gemacht? (push-Prinzip)

Figure 68: Driving

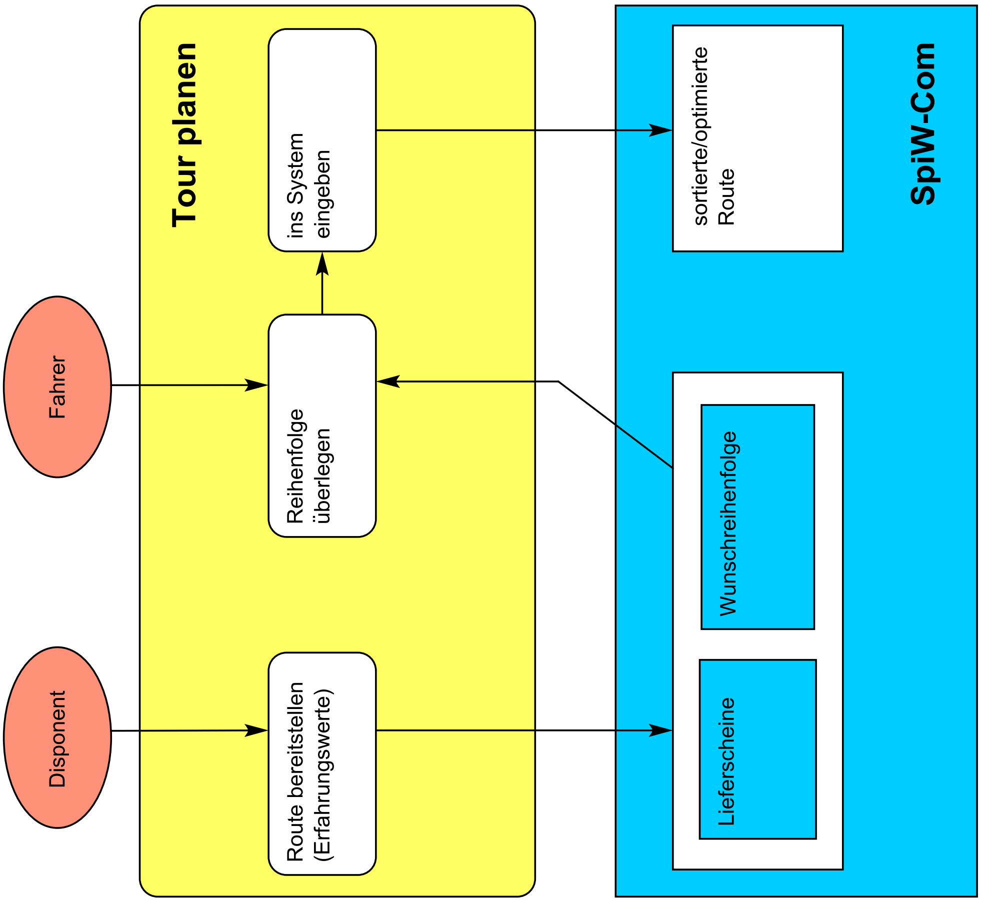


Figure 69: Planning a Tour

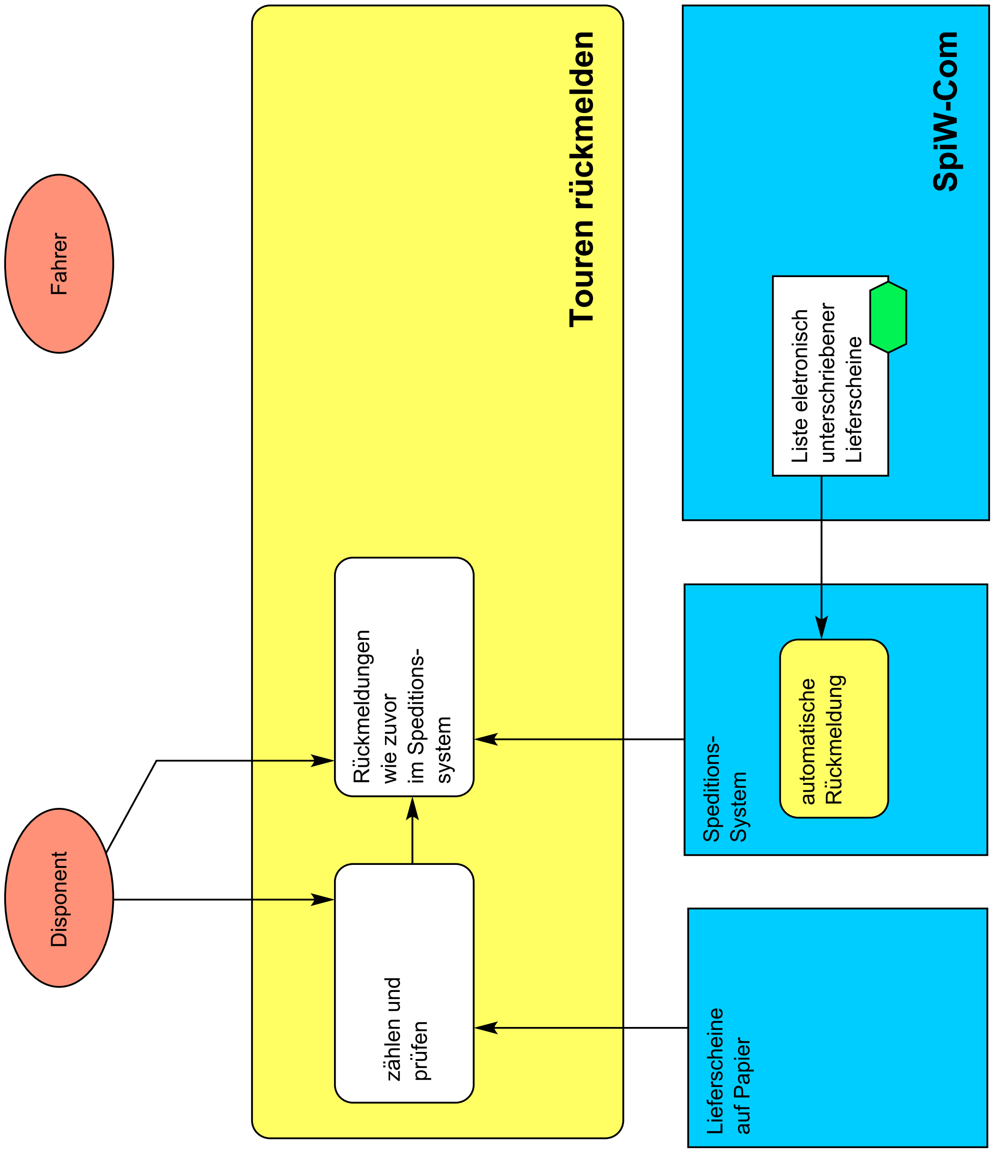


Figure 70: Report a Tour

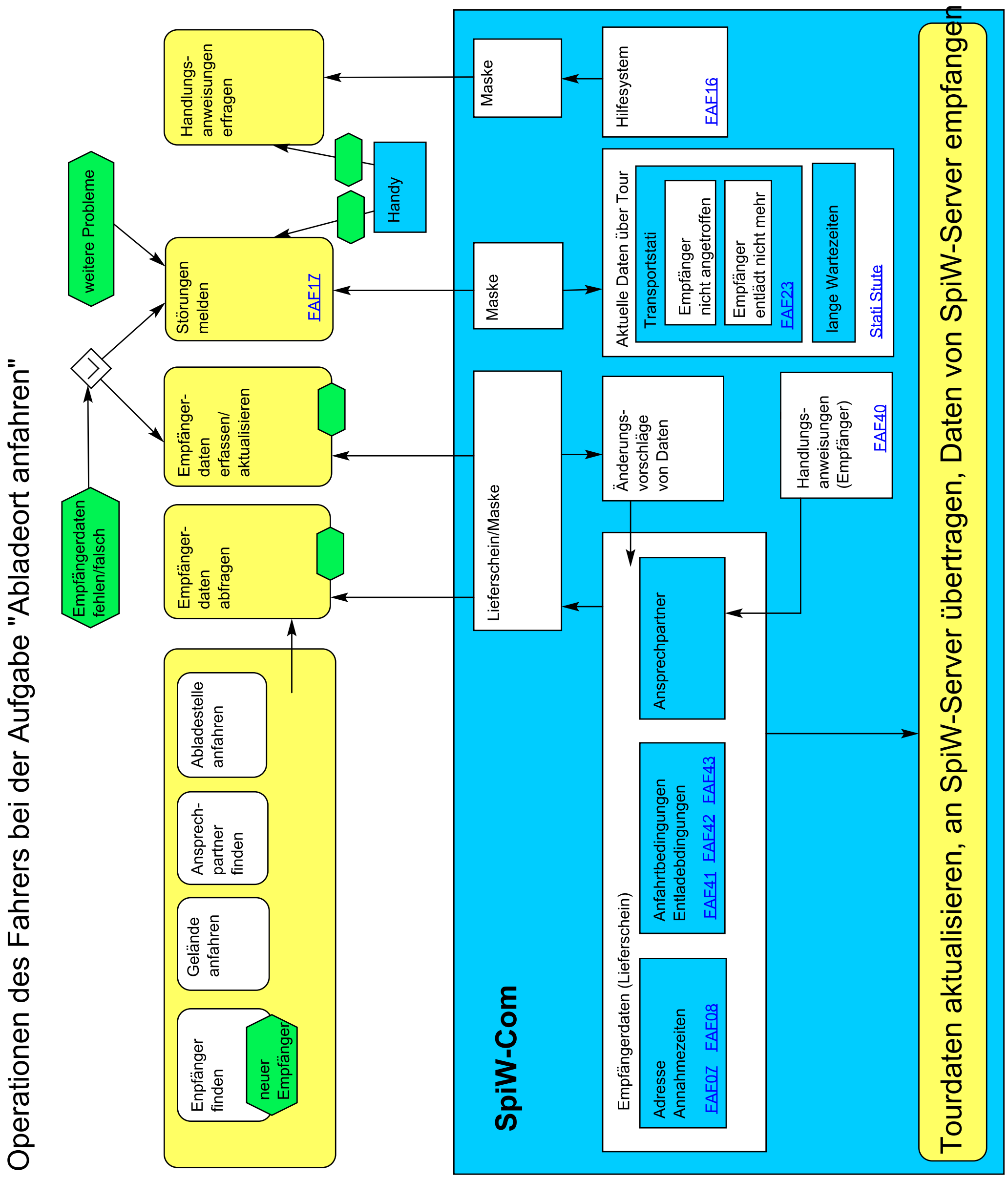

Operationen des Fahrers bei der Aufgabe "Abladeort anfahren"
Empfänger finden
neuer Empfänger
Gelände anfahren
Ansprech-partner finden
Abladestelle anfahren
Empfängerdaten fehlen/falsch
Empfänger-daten abfragen
Empfänger-daten erfassen/ aktualisieren
Störungen melden
FAF17
weitere Probleme
Handlungs-anweisungen erfragen
Handy
SpiW-Com
Lieferschein/Maske
Maske
Maske
Hilfesystem
FAF16
Aktuelle Daten über Tour
Transportstati
Empfänger nicht angetroffen
Empfänger entlädt nicht mehr
FAF23
lange Wartezeiten
Stati Stute
Änderungs-vorschläge von Daten
Handlungs-anweisungen (Empfänger)
FAF40
Empfängerdaten (Lieferschein)
Adresse Annahmezeiten
FAF07 FAF08
Anfahrtbedingungen Entladebdingungen
FAF41 FAF42 FAF43
Ansprechpartner
Tourdaten aktualisieren, an SpiW-Server übertragen, Daten von SpiW-Server empfangen

Figure 71: Arrival at Customer’s Site with Detailed Description of SpiW-Com

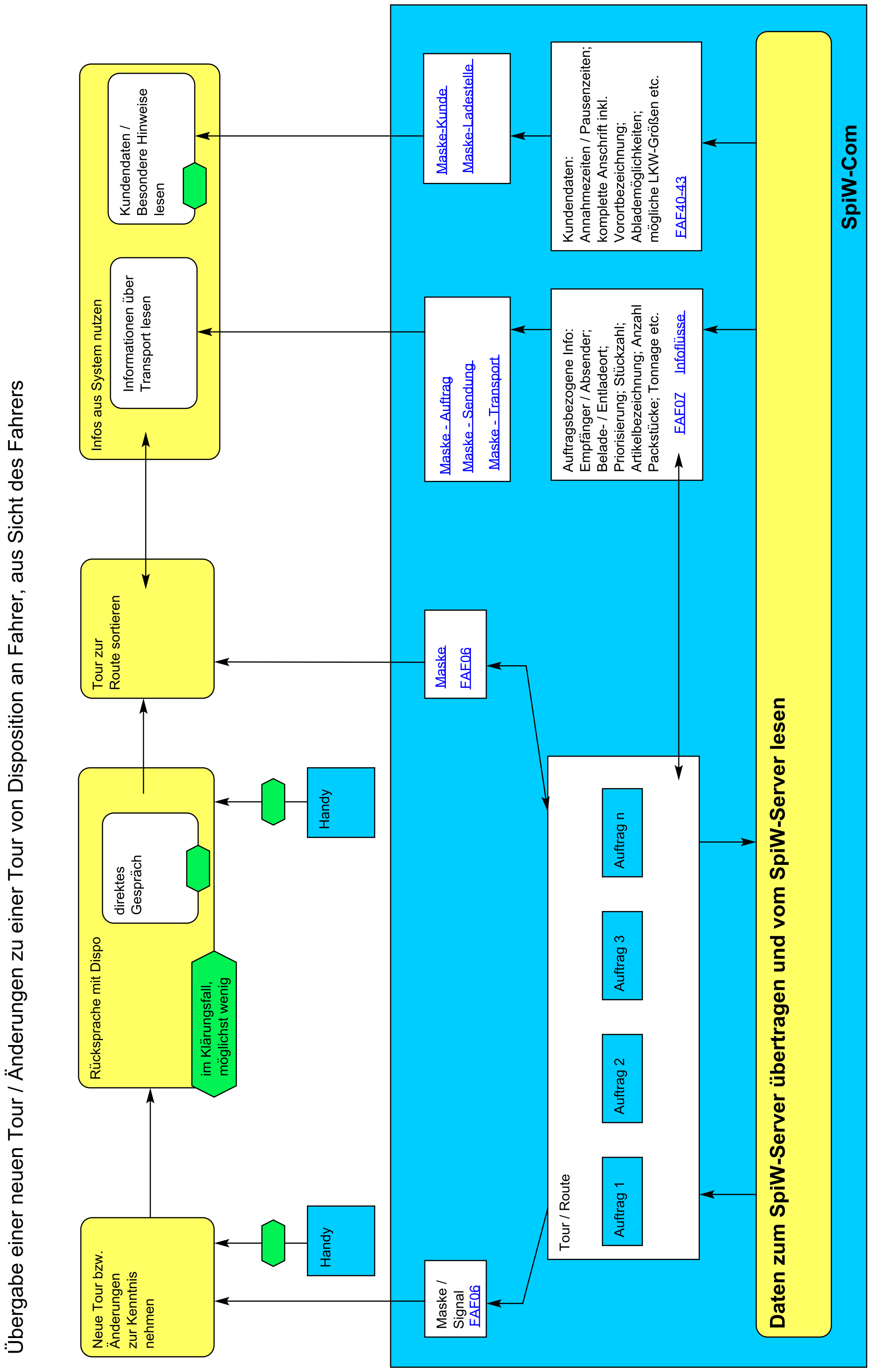

Übergabe einer neuen Tour / Änderungen zu einer Tour von Disposition an Fahrer, aus Sicht des Fahrers
Neue Tour bzw. Änderungen zur Kenntnis nehmen
Rücksprache mit Dispo
direktes Gespräch
im Klärungsfall, möglichst wenig
Tour zur Route sortieren
Infos aus System nutzen
Informationen über Transport lesen
Kundendaten / Besondere Hinweise lesen
Handy
Handy
Maske / Signal FAF06
Maske FAF06
Maske - Auftrag
Maske - Sendung
Maske - Transport
Maske-Kunde
Maske-Ladestelle
Tour / Route
Auftrag 1
Auftrag 2
Auftrag 3
Auftrag n
Auftragsbezogene Info: Empfänger / Absender; Belade- / Entladeort; Priorisierung; Stückzahl; Artikelbezeichnung; Anzahl Packstücke; Tonnage etc.
FAF07
Infoflüsse
Kundendaten: Annahmezeiten / Pausenzeiten; komplette Anschrift inkl. Vorortbezeichnung; Abladeemöglichkeiten; mögliche LKW-Größen etc.
FAF40-43
Daten zum SpiW-Server übertragen und vom SpiW-Server lesen
SpiW-Com

Figure 72: Information about a New Tour

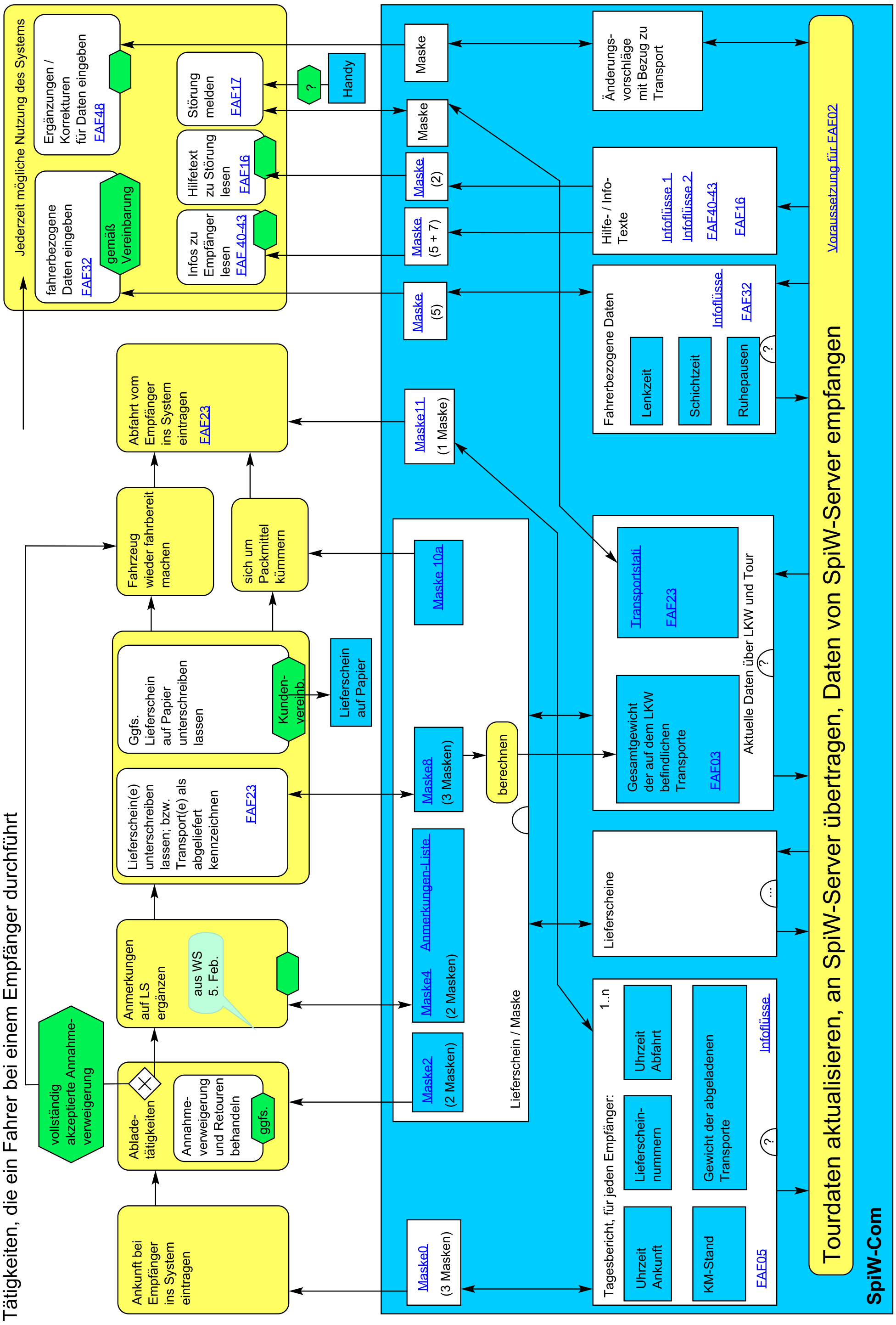

Tätigkeiten, die ein Fahrer bei einem Empfänger durchführt
Jederzeit mögliche Nutzung des Systems
fahrerbezogene Daten eingeben FAF32
gemäß Vereinbarung
Ergänzungen / Korrekturen für Daten eingeben FAF48
Infos zu Empfänger lesen FAF 40-43
Hilfetext zu Störung lesen FAF16
Störung melden FAF17
?
Handy
Ankunft bei Empfänger ins System eintragen
vollständig akzeptierte Annahme-verweigerung
Ablade-tätigkeiten
Annahme-verweigerung und Retouren behandeln
ggfs.
Anmerkungen auf LS ergänzen
aus WS 5. Feb.
Lieferschein(e) unterschreiben lassen; bzw. Transport(e) als abgeliefert kennzeichnen FAF23
Ggfs. Lieferschein auf Papier unterschreiben lassen
Kunden-vereinb.
Lieferschein auf Papier
Fahrzeug wieder fahrbereit machen
sich um Packmittel kümmern
Abfahrt vom Empfänger ins System eintragen FAF23
Maske0 (3 Masken)
Maske2 (2 Masken)
Maske4 Anmerkungen-Liste (2 Masken)
Maske8 (3 Masken)
berechnen
Maske 10a
Lieferschein / Maske
Maske11 (1 Maske)
Maske (5)
Maske (5 + 7)
Maske (2)
Maske
Maske
Tagesbericht, für jeden Empfänger: 1..n
Uhrzeit Ankunft
Lieferschein-nummern
Uhrzeit Abfahrt
KM-Stand
Gewicht der abgeladenen Transporte
FAF05
Infoflüsse
?
Lieferscheine
...
Gesamtgewicht der auf dem LKW befindlichen Transporte FAF03
Transportstati FAF23
Aktuelle Daten über LKW und Tour
?
Fahrerbezogene Daten
Lenkzeit
Schichtzeit
Ruhepausen
?
Infoflüsse
FAF32
Hilfe- / Info-Texte
Infoflüsse 1
Infoflüsse 2
FAF40-43
FAF16
Änderungs-vorschläge mit Bezug zu Transport
Tourdaten aktualisieren, an SpiW-Server übertragen, Daten von SpiW-Server empfangen
Voraussetzung für FAF02
SpiW-Com

Figure 73: Tasks at Customer’s Site with Detailed Description of SpiW-Com

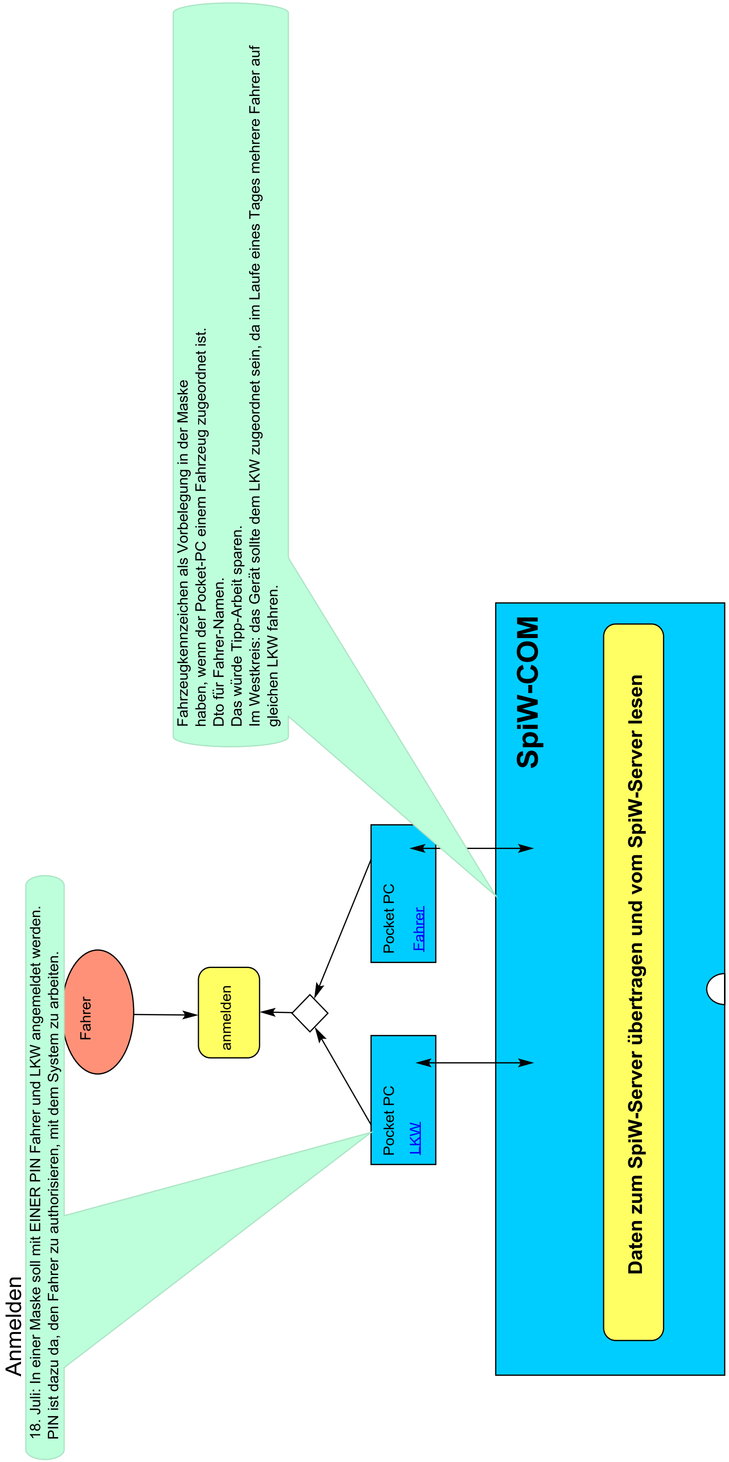

Anmelden
18. Juli: In einer Maske soll mit EINER PIN Fahrer und LKW angemeldet werden.
PIN ist dazu da, den Fahrer zu authorisieren, mit dem System zu arbeiten.
Fahrer
anmelden
Pocket PC
LKW
Pocket PC
Fahrer
Fahrzeugkennzeichen als Vorbelegung in der Maske
haben, wenn der Pocket-PC einem Fahrzeug zugeordnet ist.
Dto für Fahrer-Namen.
Das würde Tipp-Arbeit sparen.
Im Westkreis: das Gerät sollte dem LKW zugeordnet sein, da im Laufe eines Tages mehrere Fahrer auf
gleichen LKW fahren.
SpiW-COM
Daten zum SpiW-Server übertragen und vom SpiW-Server lesen

Figure 74: Login (with comments added during STWT-3)

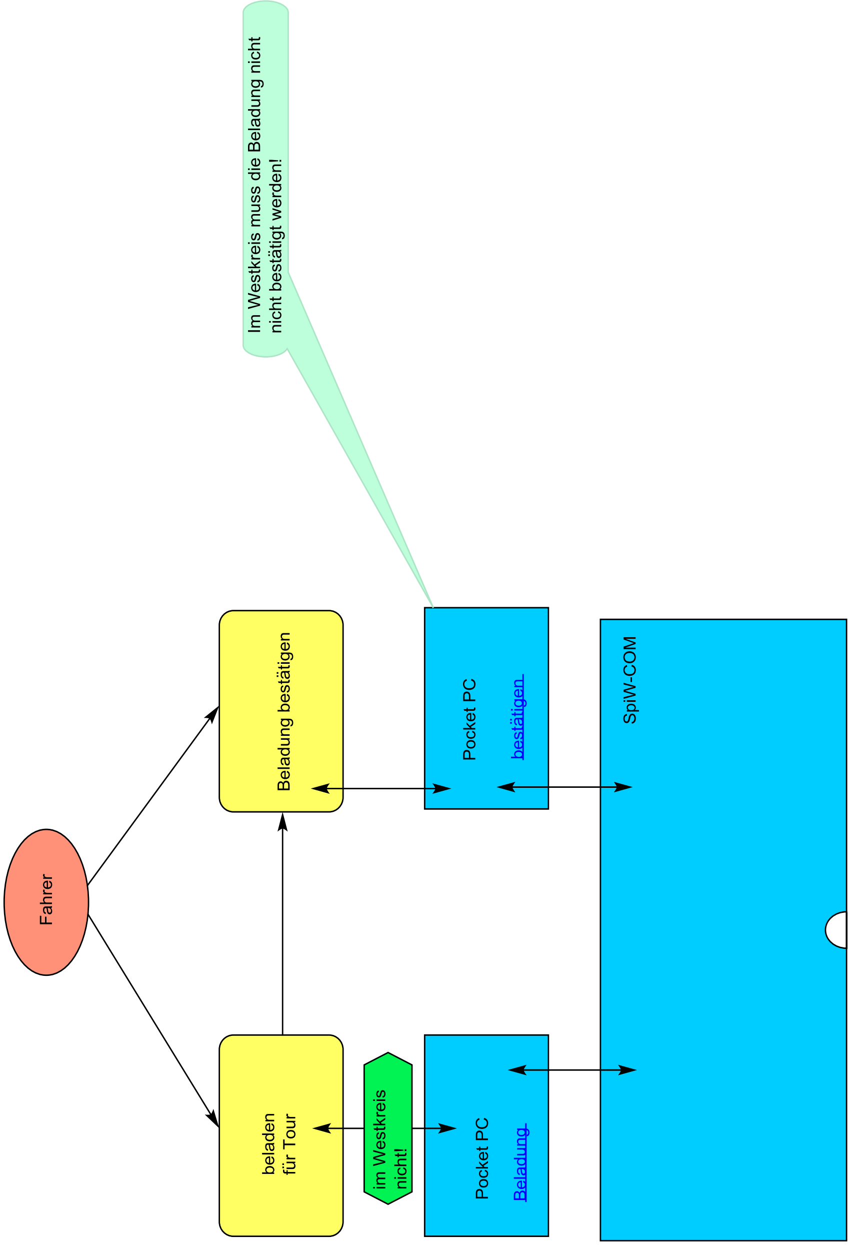


Figure 75: Loading the Trucks

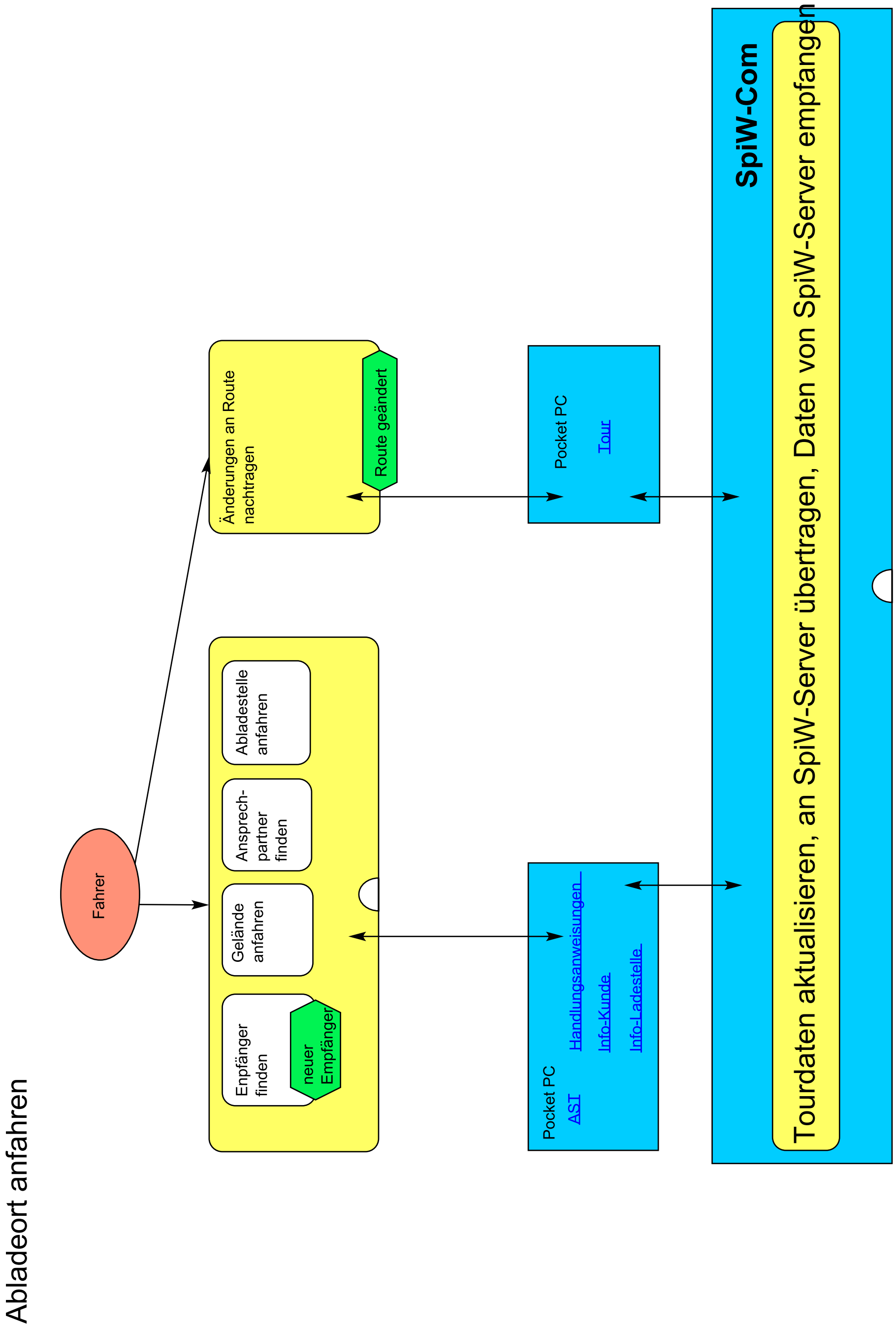


Figure 76: Arrival at a Customer's Site

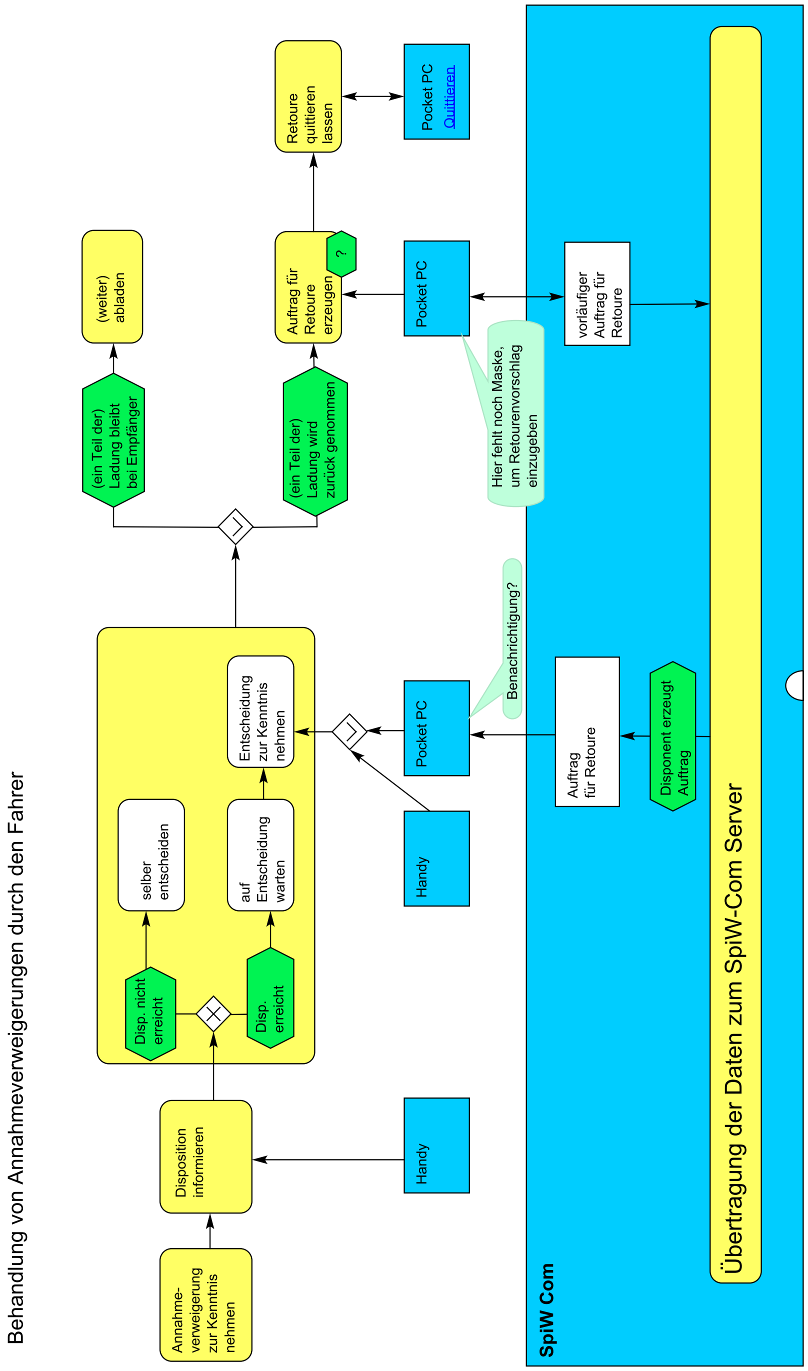

Behandlung von Annahmeverweigerungen durch den Fahrer
Annahme-verweigerung zur Kenntnis nehmen
Disposition informieren
Handy
Disp. nicht erreicht
Disp. erreicht
selber entscheiden
auf Entscheidung warten
Entscheidung zur Kenntnis nehmen
Handy
Pocket PC
Benachrichtigung?
(ein Teil der) Ladung bleibt bei Empfänger
(weiter) abladen
(ein Teil der) Ladung wird zurück genommen
Auftrag für Retoure erzeugen
?
Retoure quittieren lassen
Pocket PC
Pocket PC Quittieren
Hier fehlt noch Maske, um Retourenvorschlag einzugeben
Auftrag für Retoure
Disponent erzeugt Auftrag
vorläufiger Auftrag für Retoure
Übertragung der Daten zum SpiW-Com Server
SpiW Com

Figure 77: Procedure in Case of Non-Acceptance

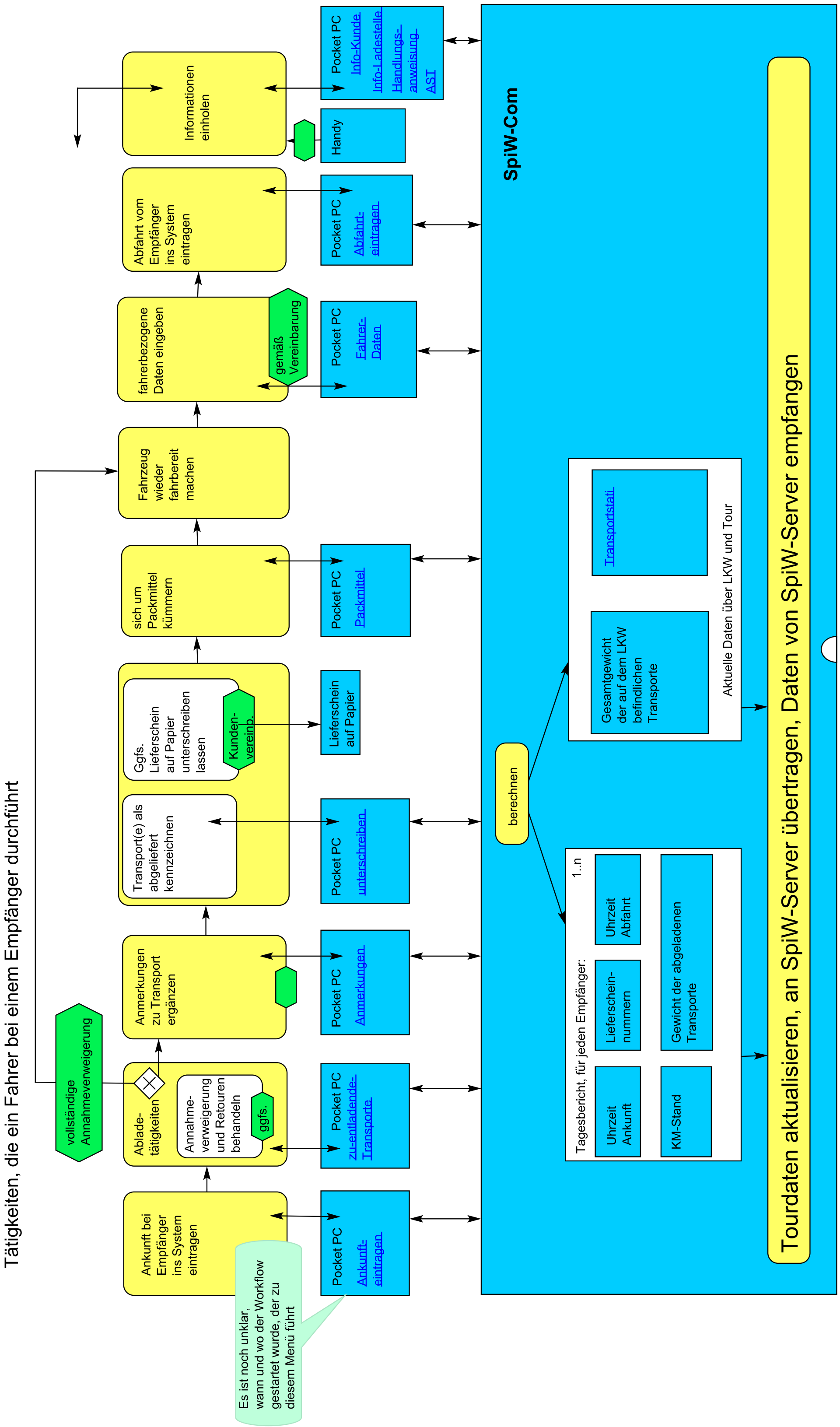
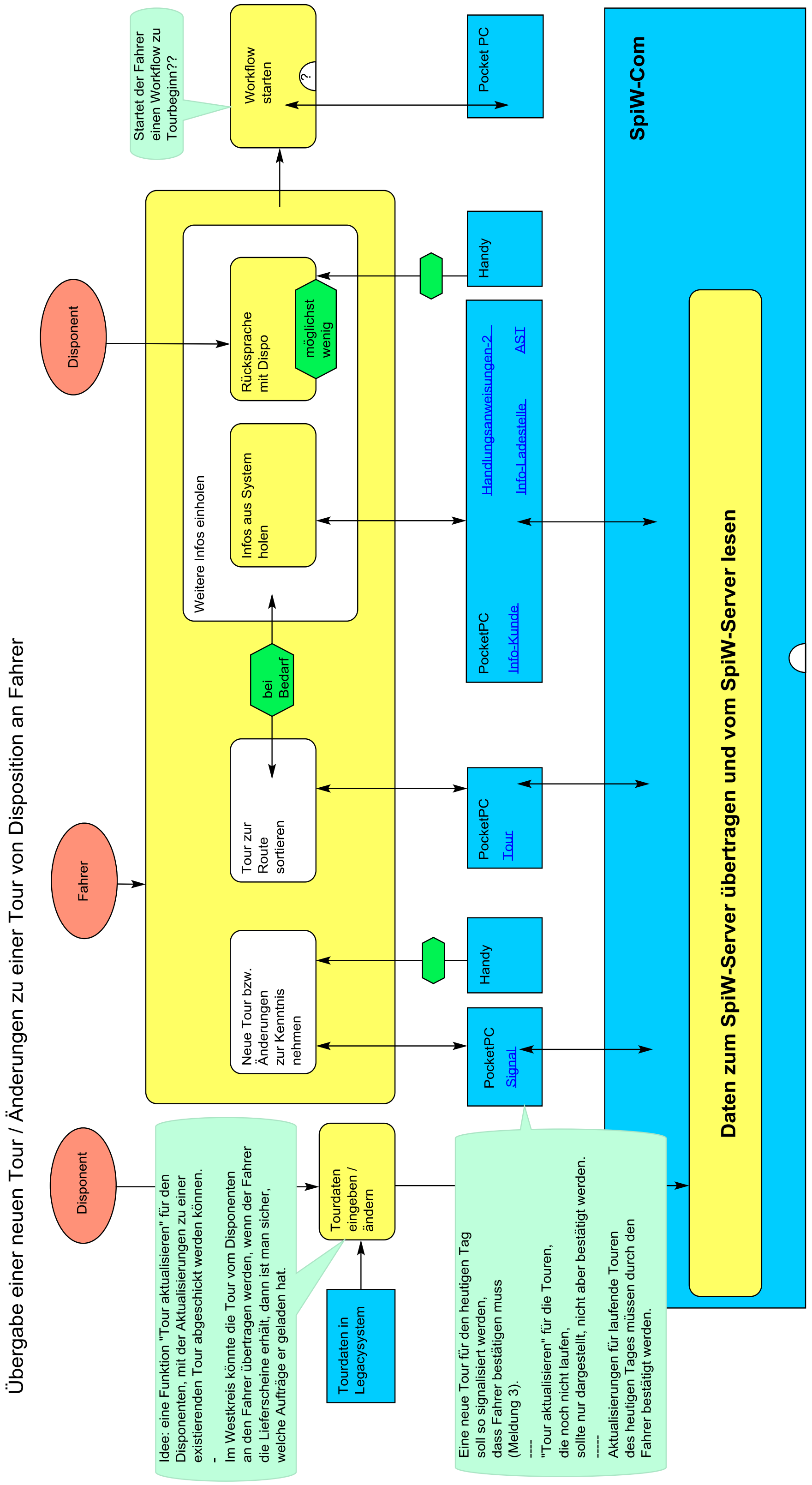

Tätigkeiten, die ein Fahrer bei einem Empfänger durchführt
vollständige Annahmeverweigerung
Ankunft bei Empfänger ins System eintragen
Ablade-tätigkeiten
Annahme-verweigerung und Retouren behandeln
ggfs.
Anmerkungen zu Transport ergänzen
Transport(e) als abgeliefert kennzeichnen
Ggfs. Lieferschein auf Papier unterschreiben lassen
Kunden-vereinb.
sich um Packmittel kümmern
Fahrzeug wieder fahrbereit machen
fahrerbezogene Daten eingeben
gemäß Vereinbarung
Abfahrt vom Empfänger ins System eintragen
Informationen einholen
Es ist noch unklar, wann und wo der Workflow gestartet wurde, der zu diesem Menü führt
Pocket PC Ankunft-eintragen
Pocket PC zu-entladende-Transporte
Pocket PC Anmerkungen
Pocket PC unterschreiben
Lieferschein auf Papier
Pocket PC Packmittel
Pocket PC Fahrer-Daten
Pocket PC Abfahrt-eintragen
Handy
Pocket PC Info-Kunde Info-Ladestelle Handlungs-anweisung AST
SpiW-Com
berechnen
Tagesbericht, für jeden Empfänger:
1..n
Uhrzeit Ankunft
Lieferschein-nummern
Uhrzeit Abfahrt
KM-Stand
Gewicht der abgeladenen Transporte
Gesamtgewicht der auf dem LKW befindlichen Transporte
Transportstati
Aktuelle Daten über LKW und Tour
Tourdaten aktualisieren, an SpiW-Server übertragen, Daten von SpiW-Server empfangen

Figure 78: Tasks to Do at the Recipient's Site



Idee: eine Funktion "Tour aktualisieren" für den Disponenten, mit der Aktualisierungen zu einer existierenden Tour abgeschickt werden können.
-
Im Westkreis könnte die Tour vom Disponenten an den Fahrer übertragen werden, wenn der Fahrer die Lieferscheine erhält, dann ist man sicher, welche Aufträge er geladen hat.

Eine neue Tour für den heutigen Tag soll so signalisiert werden, dass Fahrer bestätigen muss (Meldung 3).

"Tour aktualisieren" für die Touren, die noch nicht laufen, sollte nur dargestellt, nicht aber bestätigt werden.

Aktualisierungen für laufende Touren des heutigen Tages müssen durch den Fahrer bestätigt werden.

Startet der Fahrer einen Workflow zu Tourbeginn??

Figure 79: Hand-over of a New Tour between Dispatcher and Driver

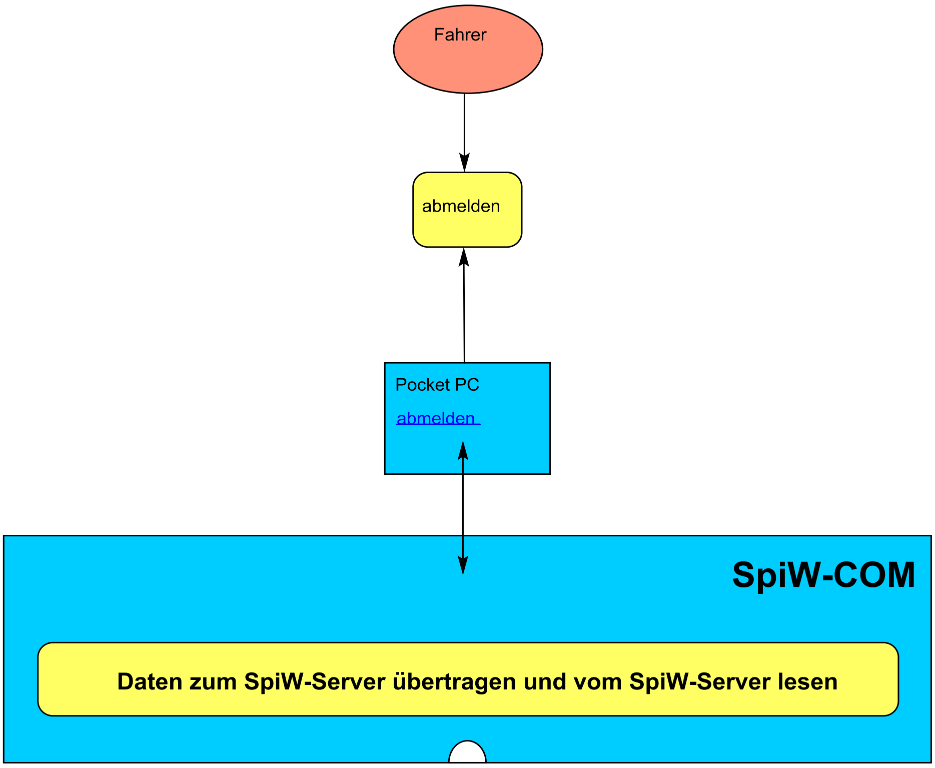


Figure 80: Logout

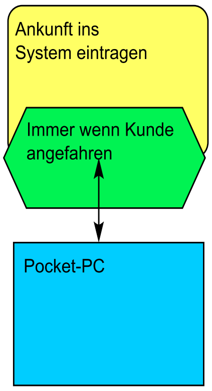


Figure 81: Document Arrival at Recipient's Site

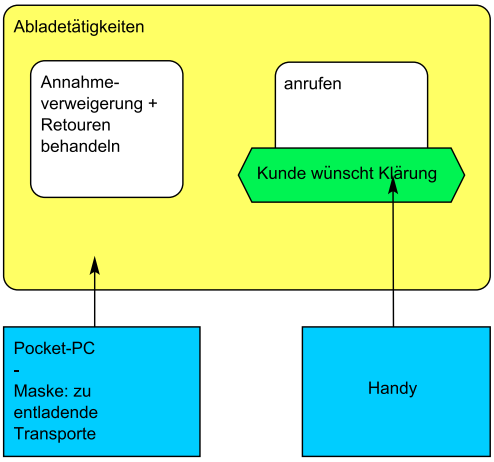

Abladetätigkeiten
Annahme-verweigerung + Retouren behandeln
anrufen
Kunde wünscht Klärung
Pocket-PC
-
Maske: zu entladende Transporte
Handy

Figure 82: Use Mobile Phone if Recipient Needs Clarification

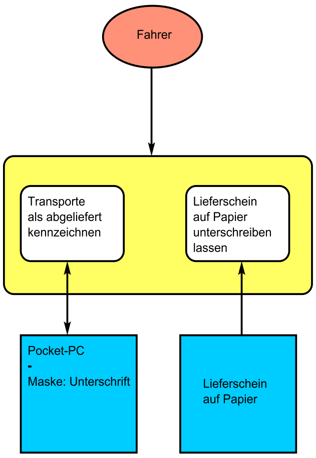

Fahrer
Transporte als abgeliefert kennzeichnen
Lieferschein auf Papier unterschreiben lassen
Pocket-PC
-
Maske: Unterschrift
Lieferschein auf Papier

Figure 83: Use Paper Form in Addition to SpiW-Com

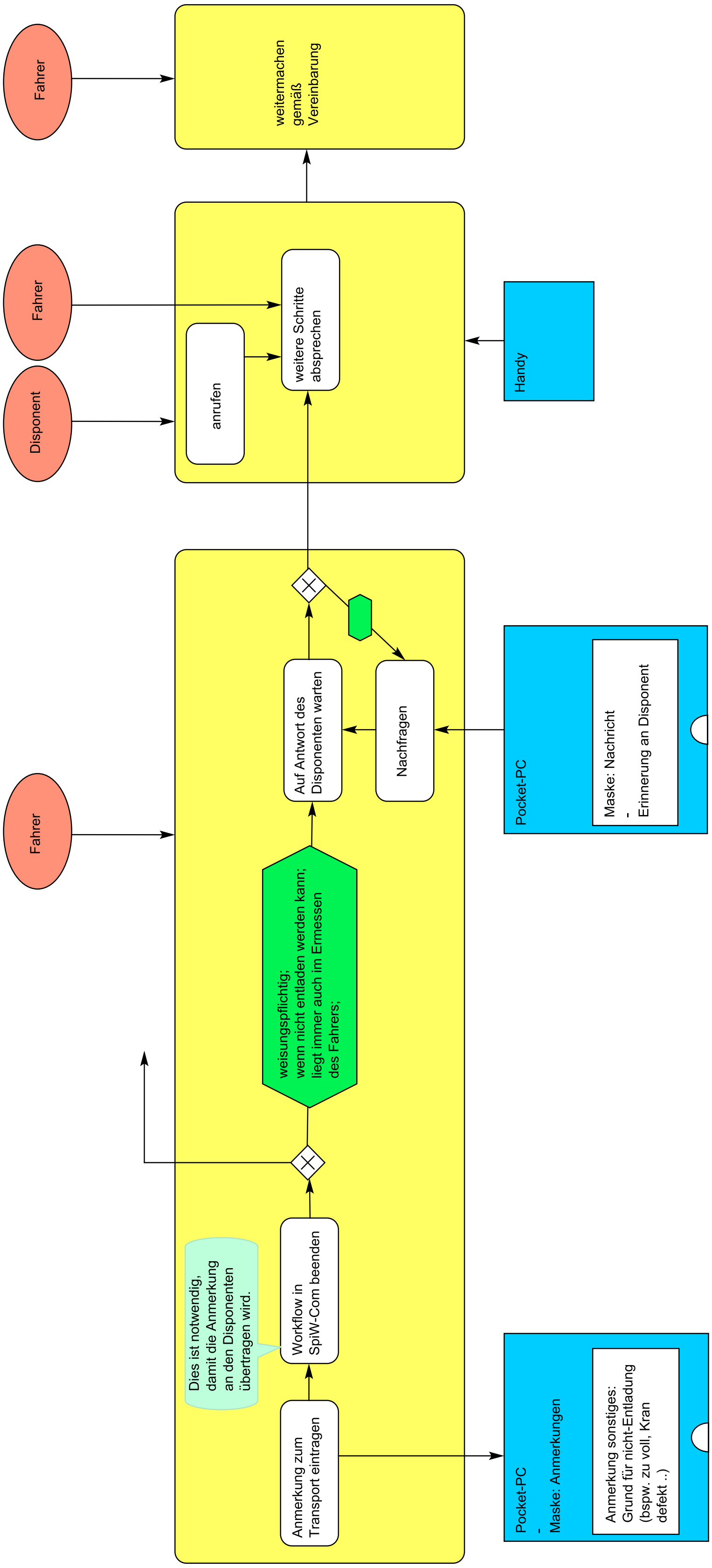
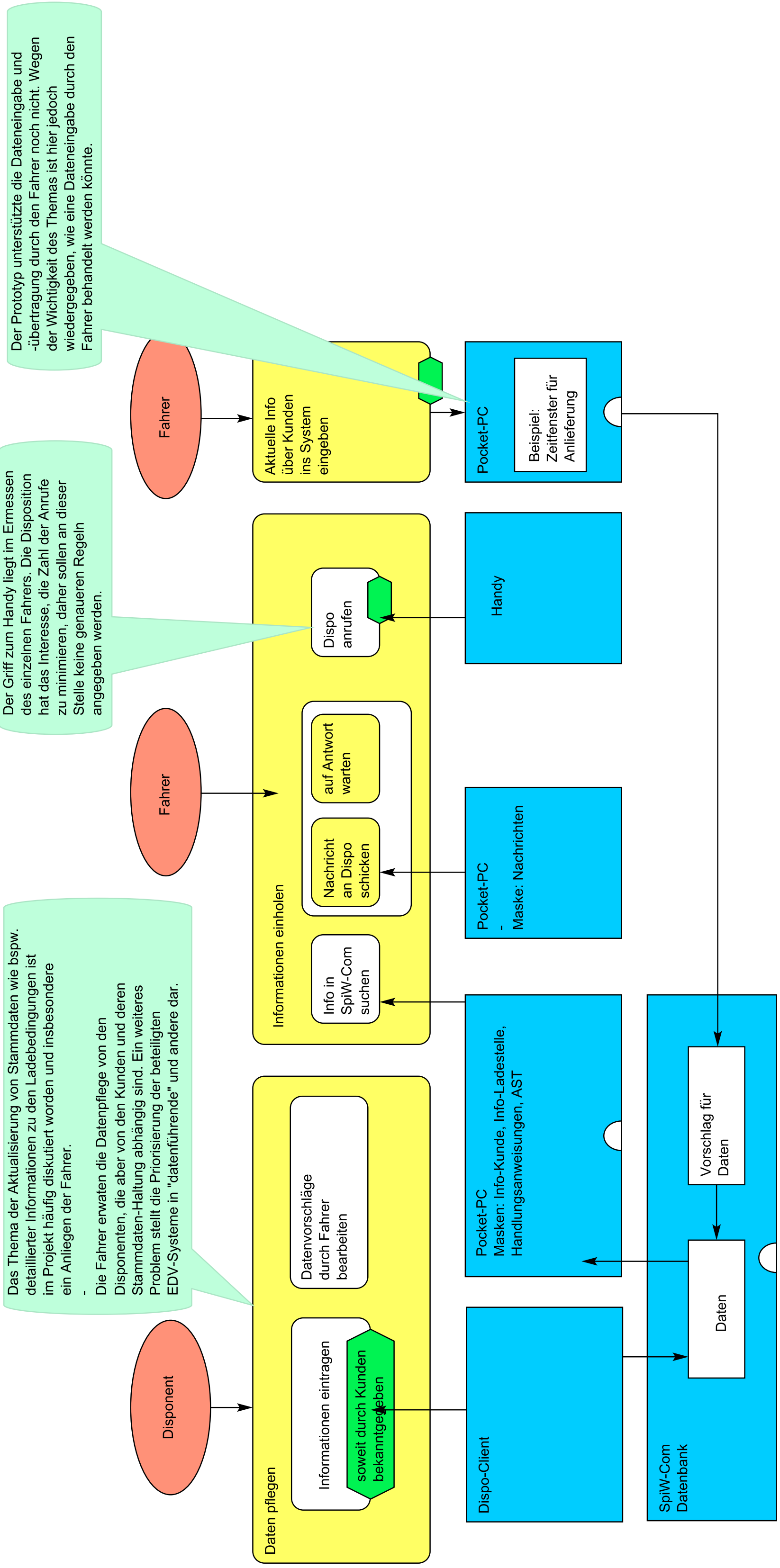


Figure 84: When must the driver Wait for further Instructions?



Das Thema der Aktualisierung von Stammdaten wie bspw. detaillierter Informationen zu den Ladebedingungen ist im Projekt häufig diskutiert worden und insbesondere ein Anliegen der Fahrer.
-
Die Fahrer erwaten die Datenpflege von den Disponenten, die aber von den Kunden und deren Stammdaten-Haltung abhängig sind. Ein weiteres Problem stellt die Priorisierung der beteiligten EDV-Systeme in "datenführende" und andere dar.

Der Griff zum Handy liegt im Ermessen des einzelnen Fahrers. Die Disposition hat das Interesse, die Zahl der Anrufe zu minimieren, daher sollen an dieser Stelle keine genaueren Regeln angegeben werden.

Der Prototyp unterstützte die Dateneingabe und -übertragung durch den Fahrer noch nicht. Wegen der Wichtigkeit des Themas ist hier jedoch wiedergegeben, wie eine Dateneingabe durch den Fahrer behandelt werden könnte.

Disponent
Daten pflegen
Informationen eintragen
soweit durch Kunden bekanntgegeben
Datenvorschläge durch Fahrer bearbeiten
Dispo-Client
SpiW-Com Datenbank
Daten
Vorschlag für Daten
Pocket-PC
Masken: Info-Kunde, Info-Ladestelle, Handlungsanweisungen, AST
Fahrer
Informationen einholen
Info in SpiW-Com suchen
Nachricht an Dispo schicken
auf Antwort warten
Dispo anrufen
Pocket-PC
-
Maske: Nachrichten
Handy
Fahrer
Aktuelle Info über Kunden ins System eingeben
Pocket-PC
Beispiel: Zeitfenster für Anlieferung

Figure 85: Drivers' Feedback into Database

11.3.2 „Electronic TOC-Service"

Figure	Content / Purpose
Figure 86 – Figure 91	
	These diagrams were prepared by a student working in the library; they present the status quo of the ToC-Service.
Figure 92: Idea for new Organization of ToC Service	
	This diagram was input into STWT-2; it sketches the team-leader's ideas of the future ToC-Service.
Figure 93 and Figure 94	
	These diagrams are the results of STWT-2 and STWT-3 in which the team elaborated technically supported work-procedures on the basis of the sketch in Figure 92.
Figure 96 – Figure 98	
	These diagrams were input into STWT-4; they describe for three technical alternatives how they would support the work-procedures.
Figure 99 – Figure 103	
	These diagrams were input into the qualification workshop. They summarize relation between CSCW-systems and -functions on the one side and tasks on the other.
Figure 104 –	
	These diagrams are results of the qualification workshop. During the workshop the participants split up into three groups, each with an assignment that resulted from the list of open issues (cp. Figure 45). The groups' results are documented in the diagrams.
Figure 107 – Figure 110	
	When the project "Electronic ToC-Service" was relaunched in Spring 2005, the group started to work with the diagram from the previous year. But feedback soon lead to changes regarding the organizational procedures as well as the CSCW-system ELISE. Two diagrams illustrate these changes as well as feedback given during two workshops.

Table 34: Overview of Diagrams in „Electronic ToC-Service"

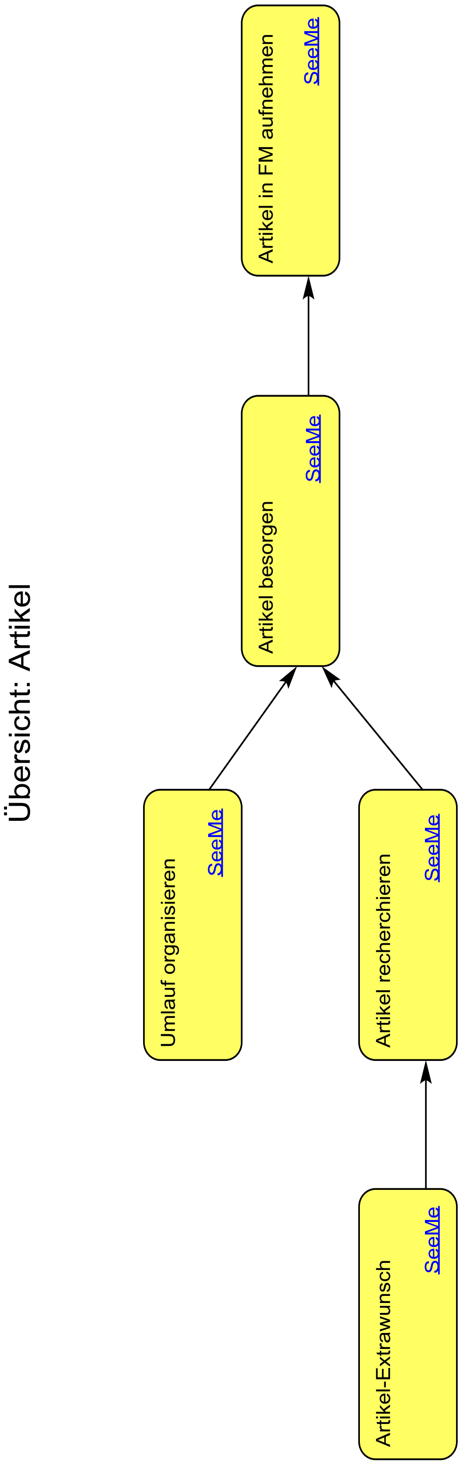


Figure 86: Overview

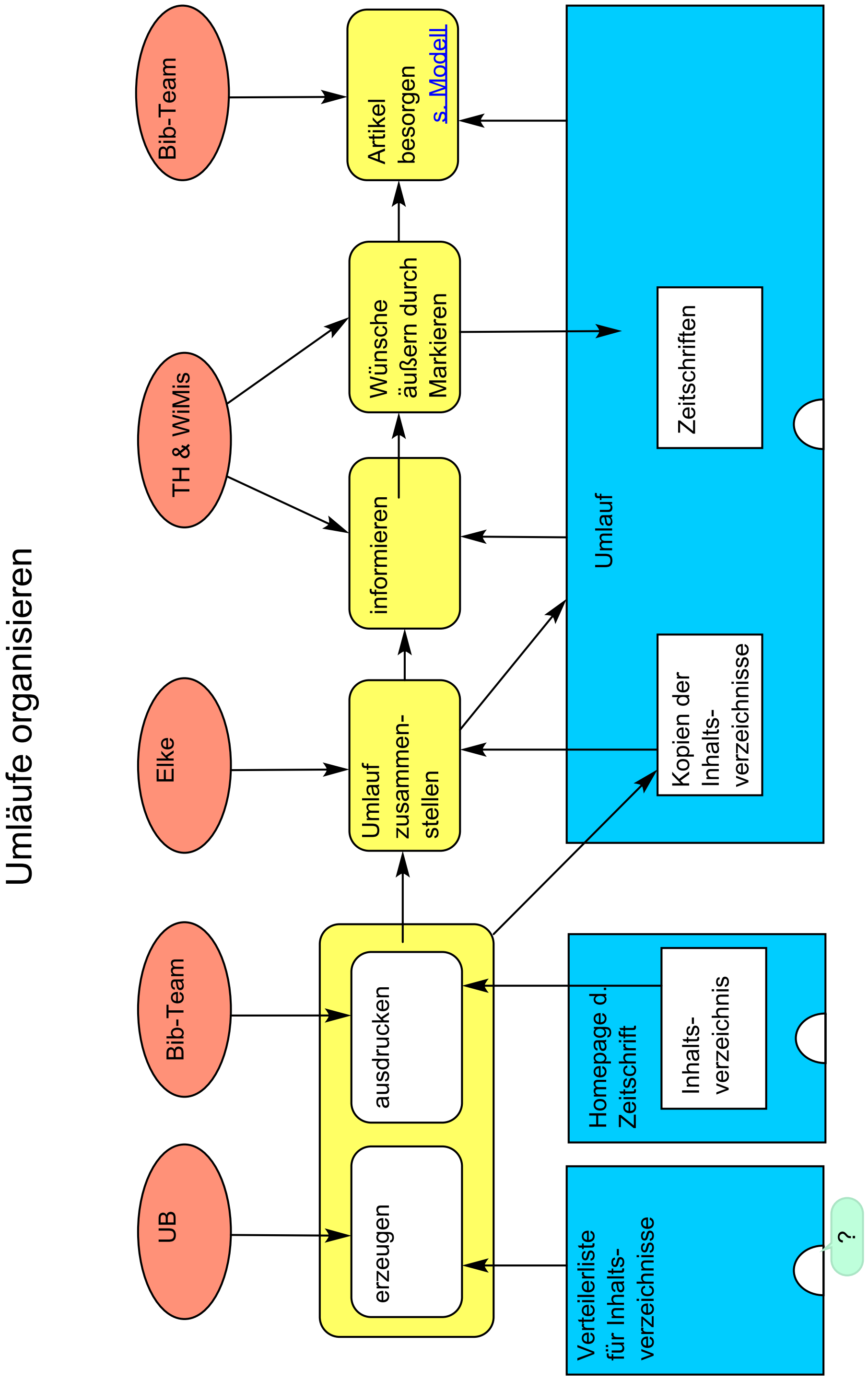


Figure 87: Organize ToC-Loop

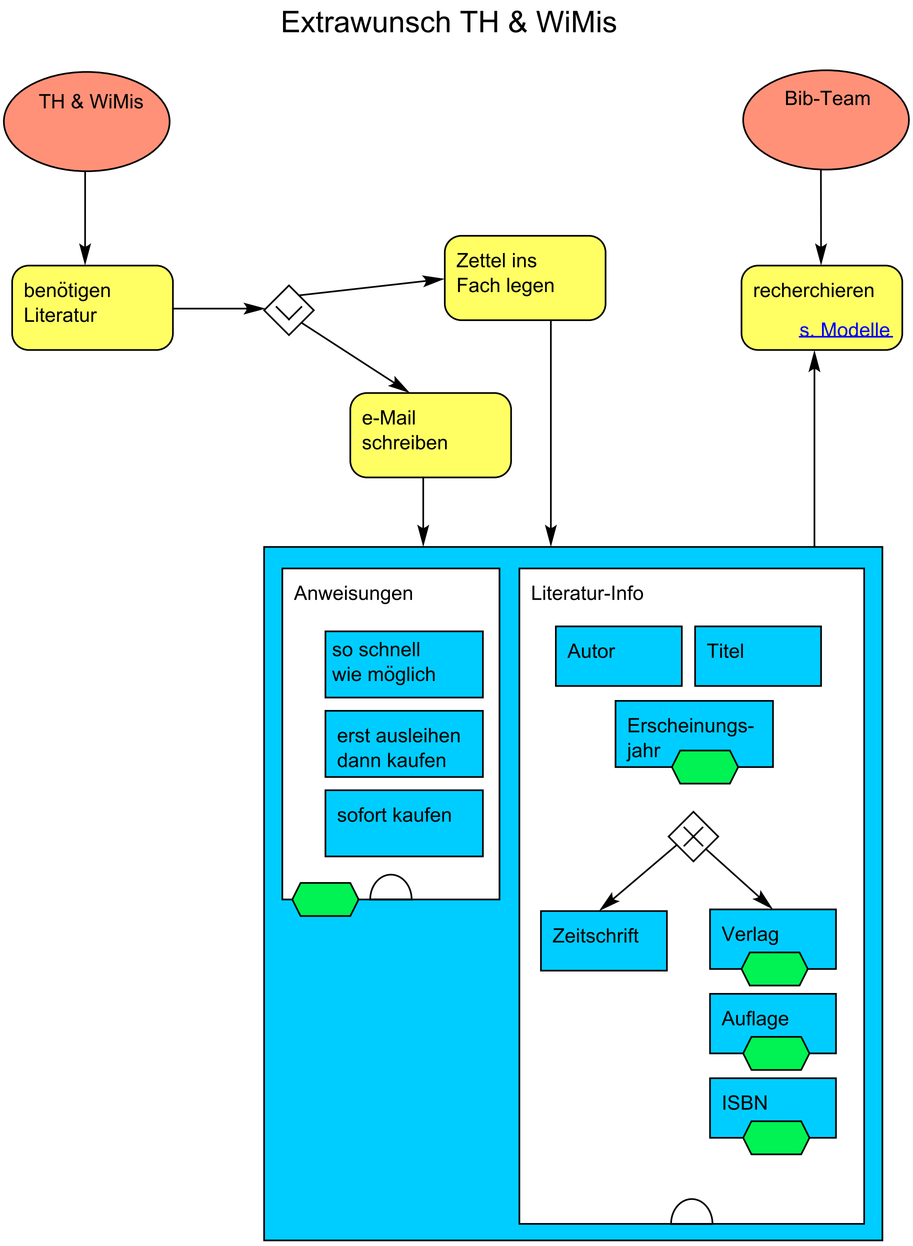


Figure 88: Special Article Order

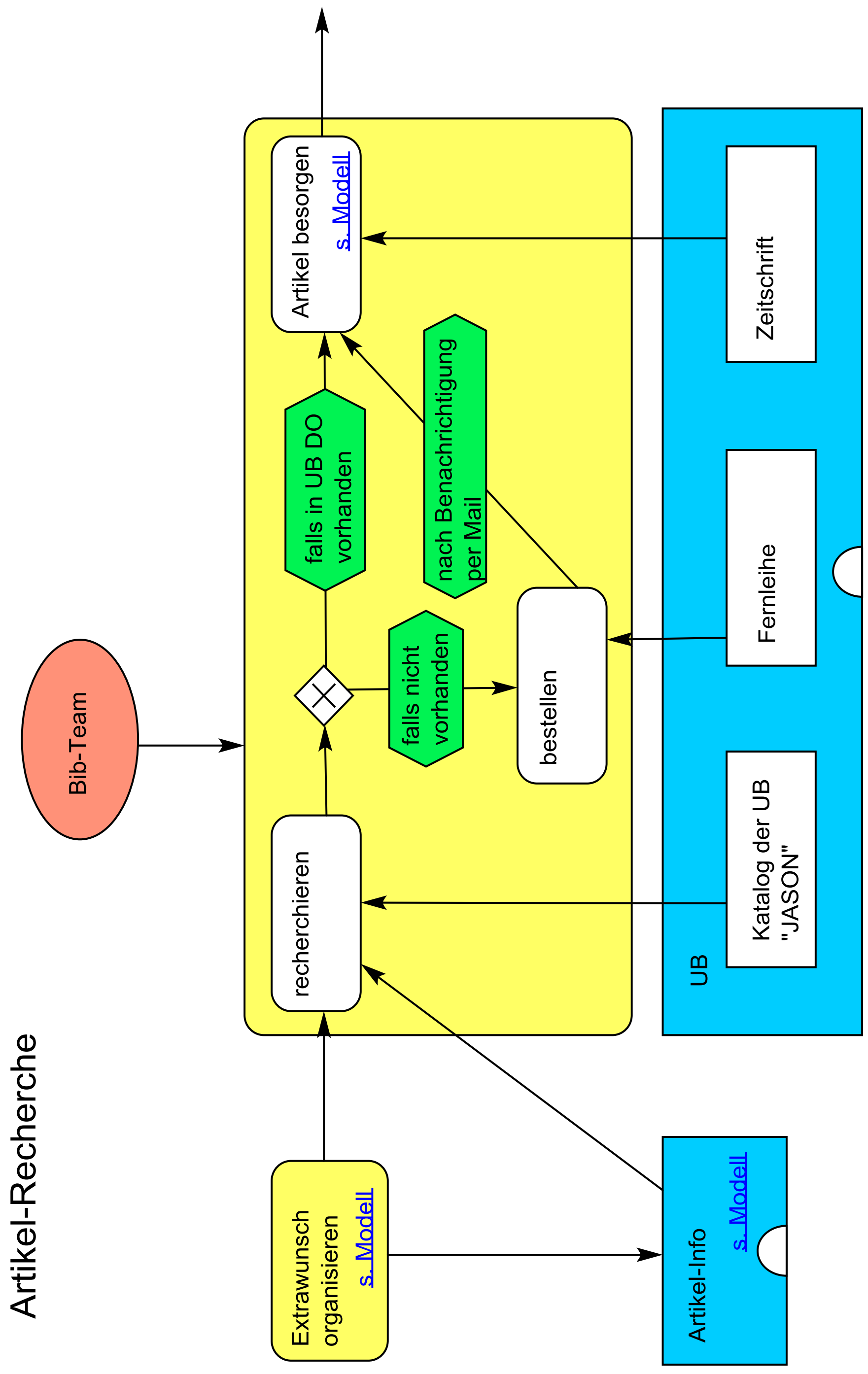
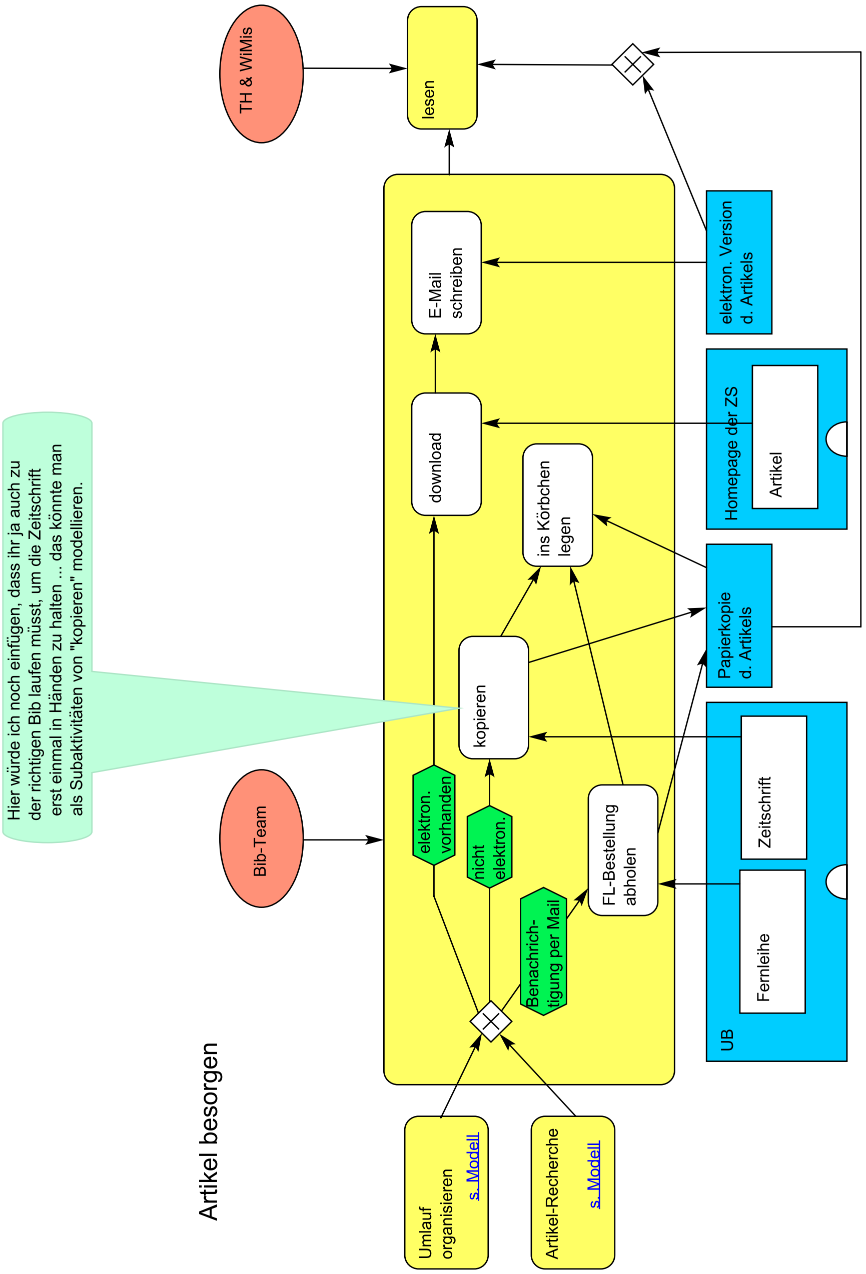


Figure 89: Search for Wanted Articles



Hier würde ich noch einfügen, dass ihr ja auch zu der richtigen Bib laufen müsst, um die Zeitschrift erst einmal in Händen zu halten ... das könnte man als Subaktivitäten von "kopieren" modellieren.

Figure 90: Get Articles

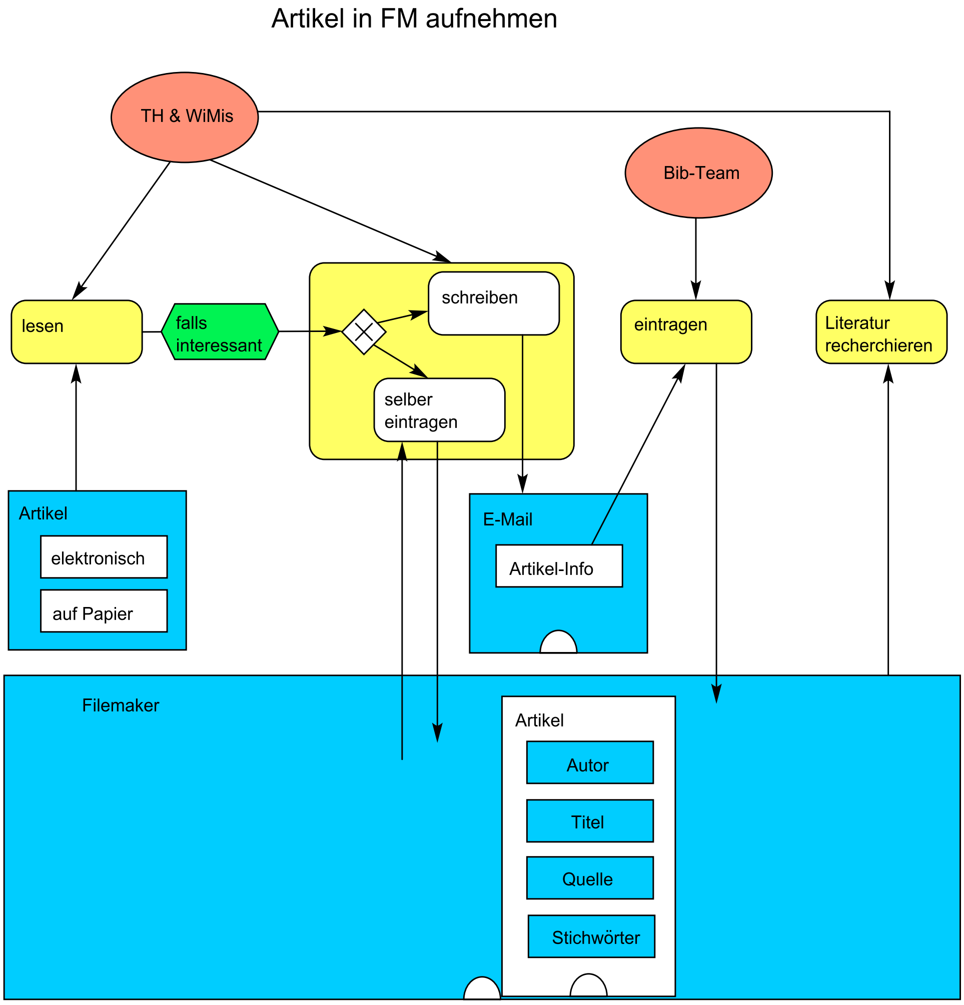


Figure 91: Register Article in Database

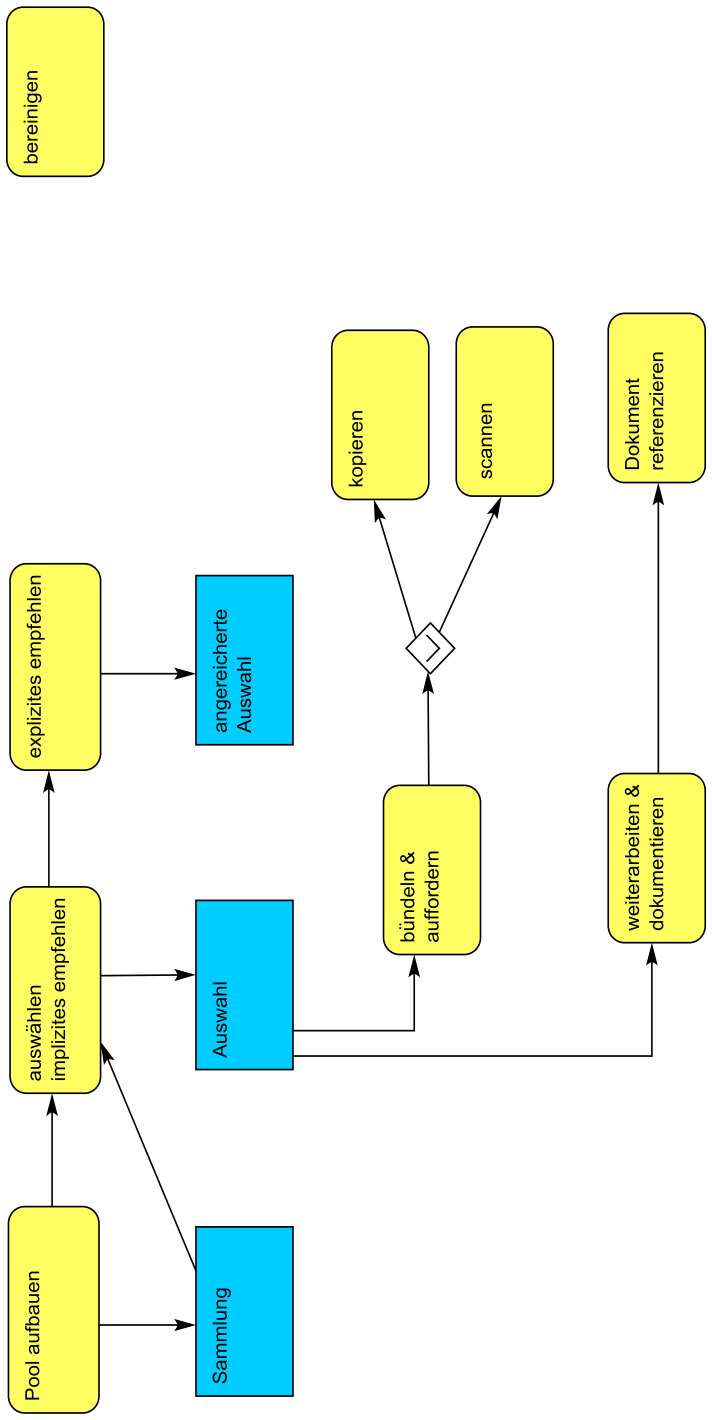


Figure 92: Idea for new Organization of ToC Service

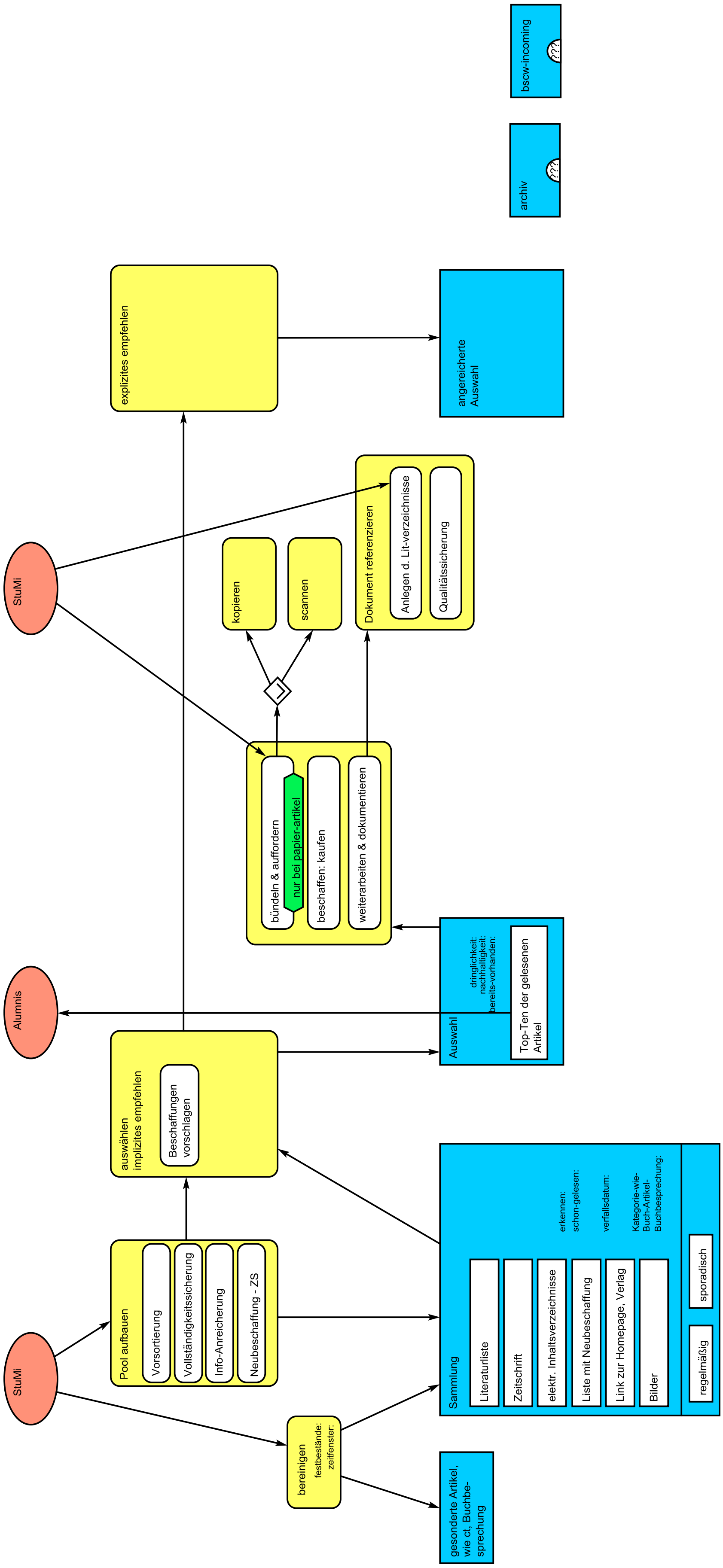


Figure 93: Result from STWT-2

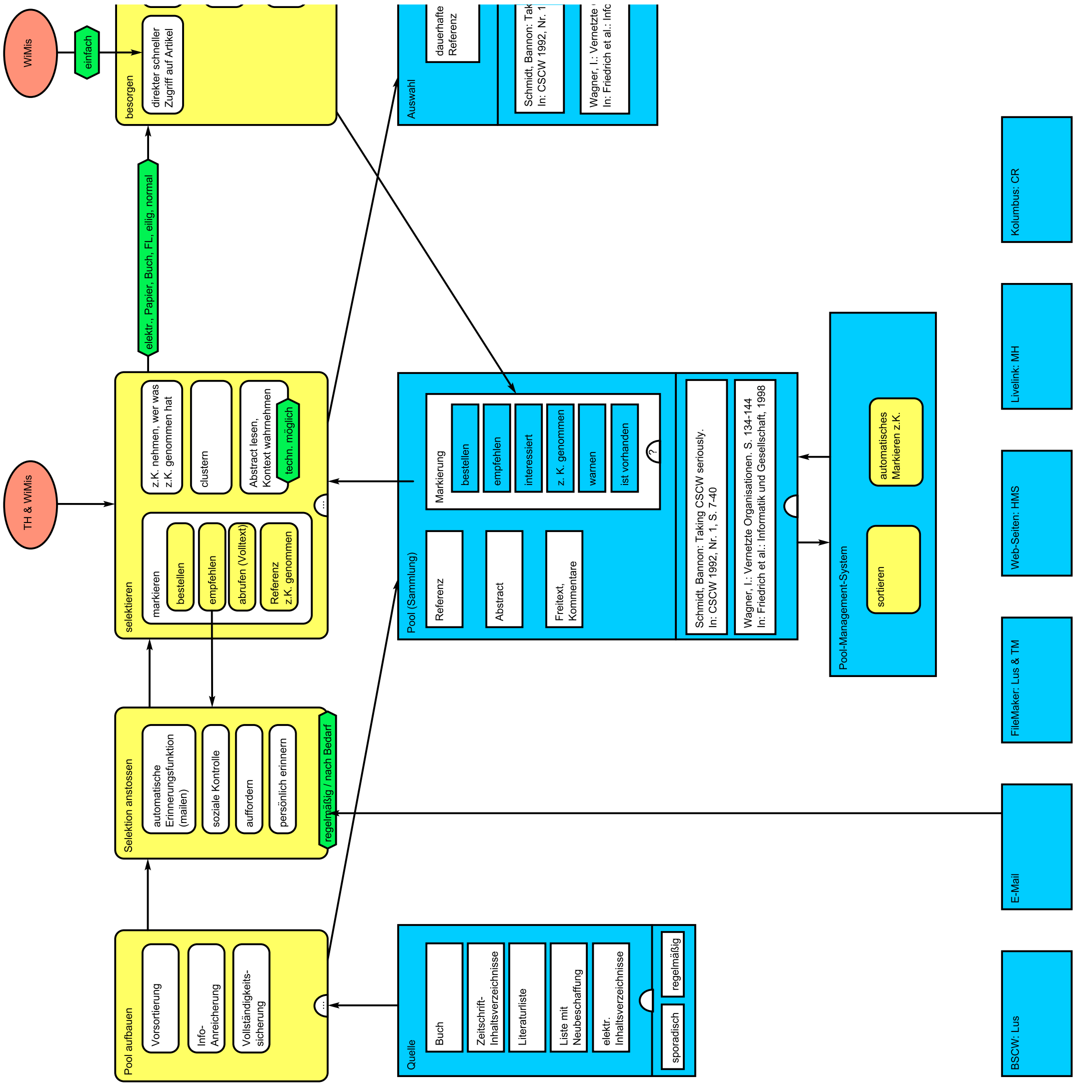


Figure 94: Result of STWT-3 (Part 1 of 2)

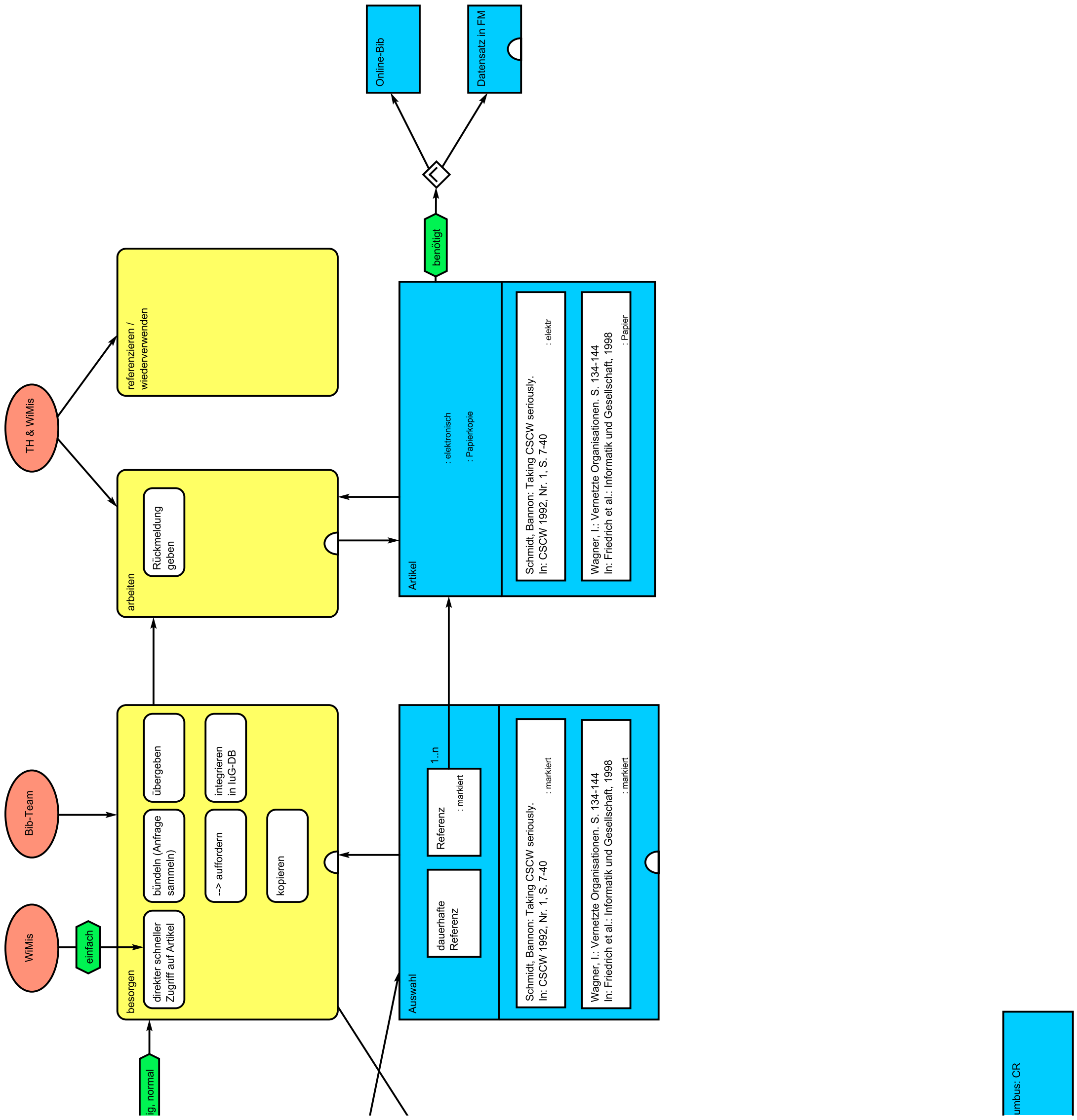

Online-Bib
Datensatz in FM
benötigt
TH & WiMis
referenzieren / wiederverwenden
arbeiten
Rückmeldung geben
Artikel
: elektronisch
: Papierkopie
Schmidt, Bannon: Taking CSCW seriously. In: CSCW 1992, Nr. 1, S. 7-40
: elektr
Wagner, I.: Vernetzte Organisationen. S. 134-144 In: Friedrich et al.: Informatik und Gesellschaft, 1998
: Papier
Bib-Team
WiMis
einfach
besorgen
direkter schneller Zugriff auf Artikel
bündeln (Anfrage sammeln)
übergeben
--> auffordern
integrieren in IuG-DB
kopieren
Auswahl
dauerhafte Referenz
Referenz
: markiert
1..n
Schmidt, Bannon: Taking CSCW seriously. In: CSCW 1992, Nr. 1, S. 7-40
: markiert
Wagner, I.: Vernetzte Organisationen. S. 134-144 In: Friedrich et al.: Informatik und Gesellschaft, 1998
: markiert
ig, normal
umbus: CR

Figure 95: Result of STWT-3 (Part 2 of 2)

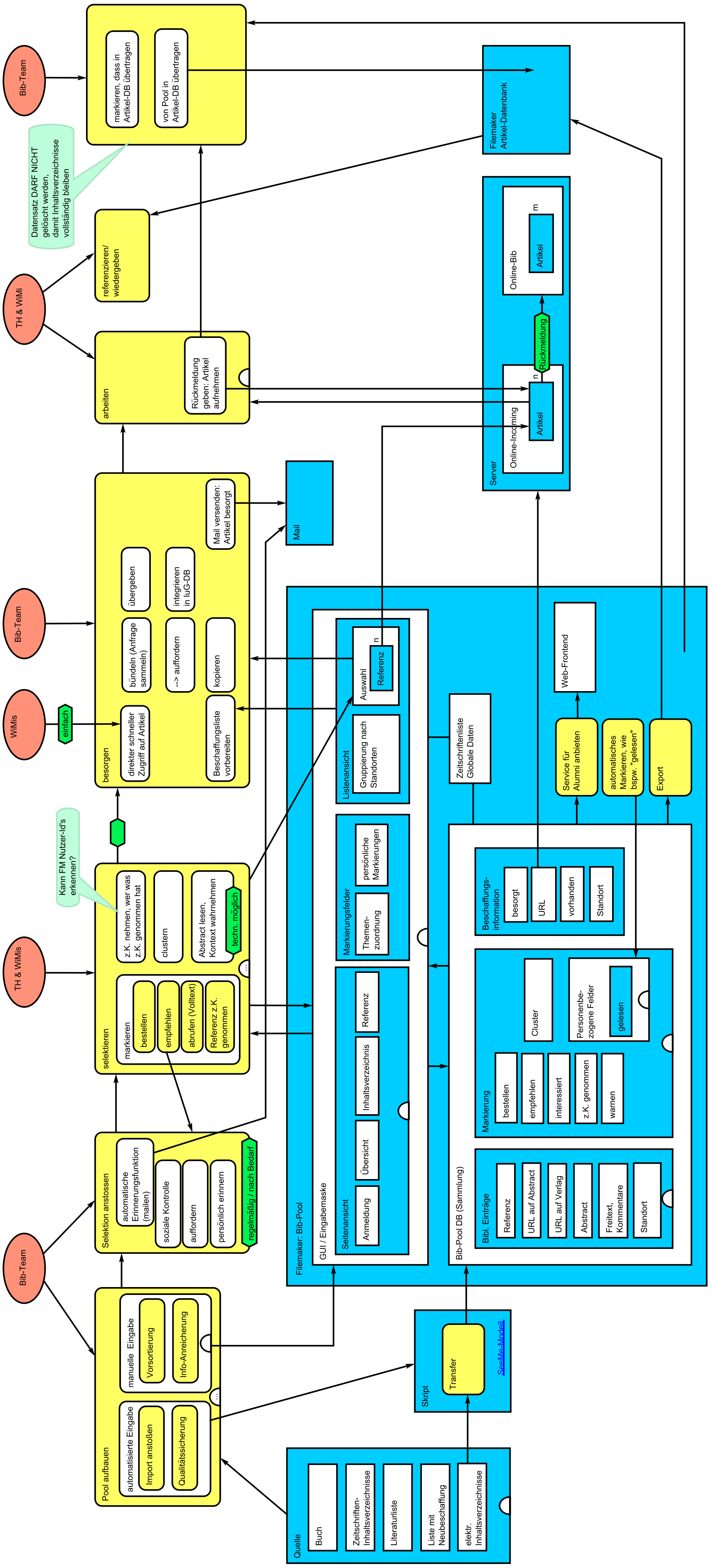
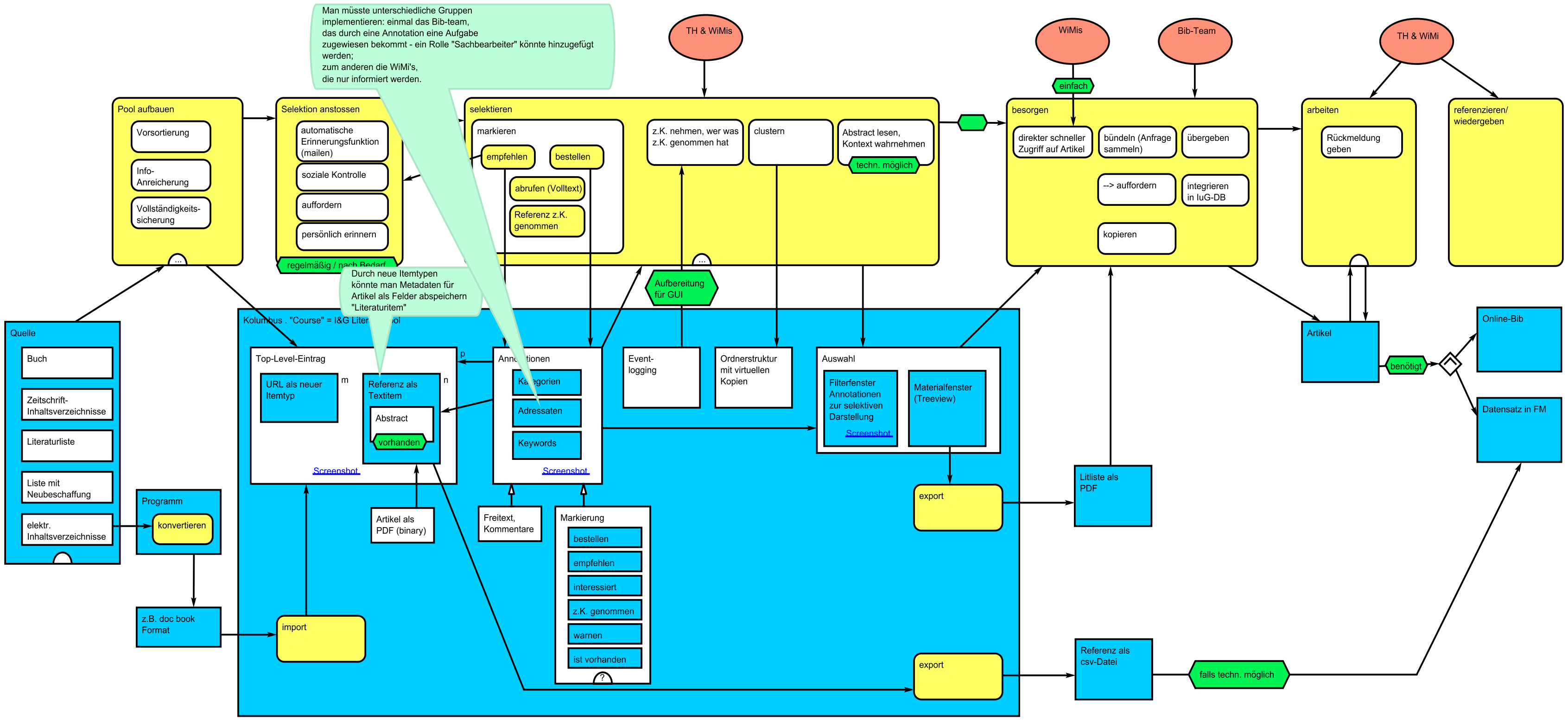


Figure 96: Technical Option Filemaker



Man müsste unterschiedliche Gruppen implementieren: einmal das Bib-team, das durch eine Annotation eine Aufgabe zugewiesen bekommt - ein Rolle "Sachbearbeiter" könnte hinzugefügt werden; zum anderen die WiMi's, die nur informiert werden.

Durch neue Itemtypen könnte man Metadaten für Artikel als Felder abspeichern "Literaturitem"

Figure 97: Technical Option K2

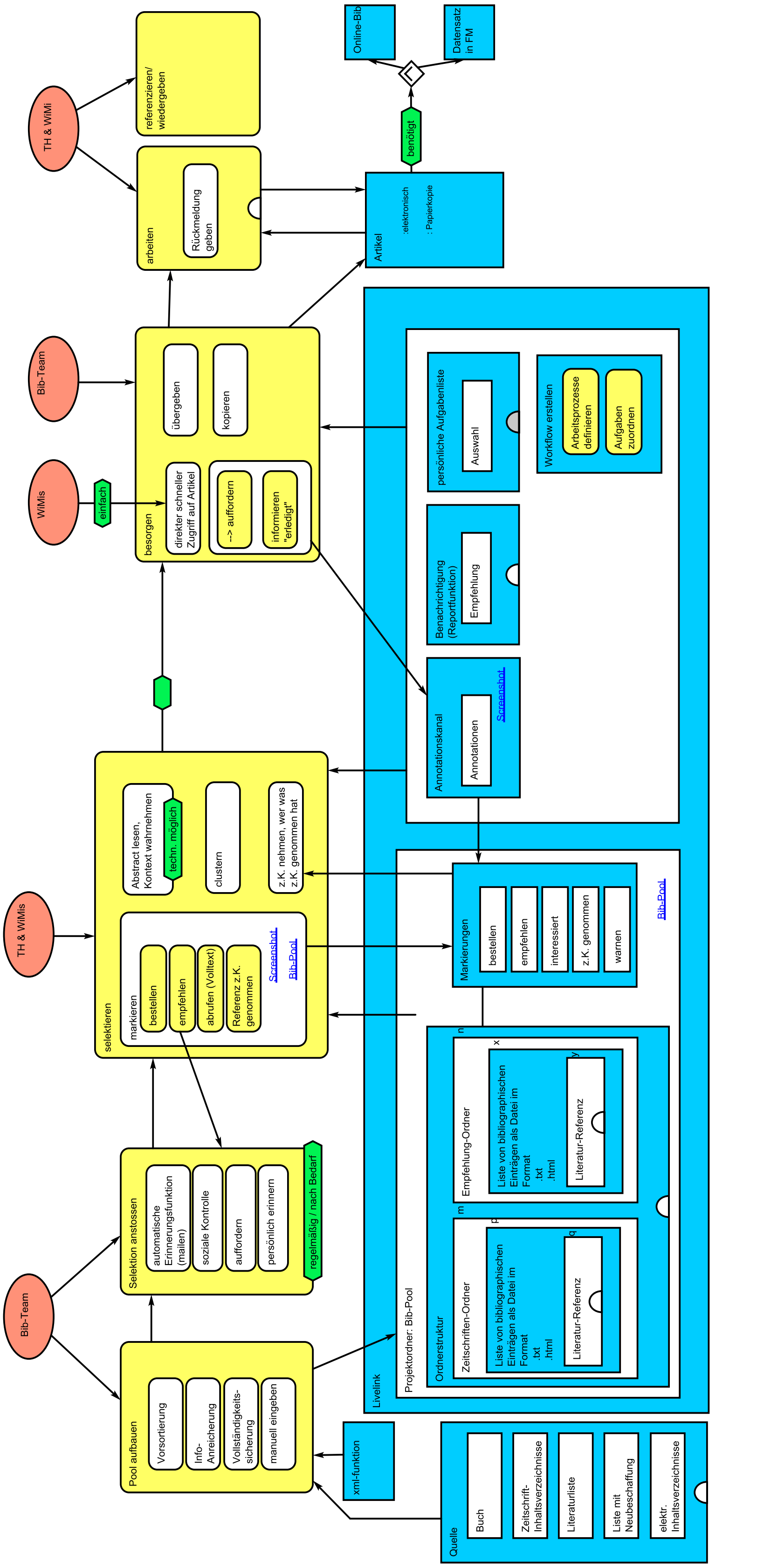


Figure 98: Technical Option LiveLink ™

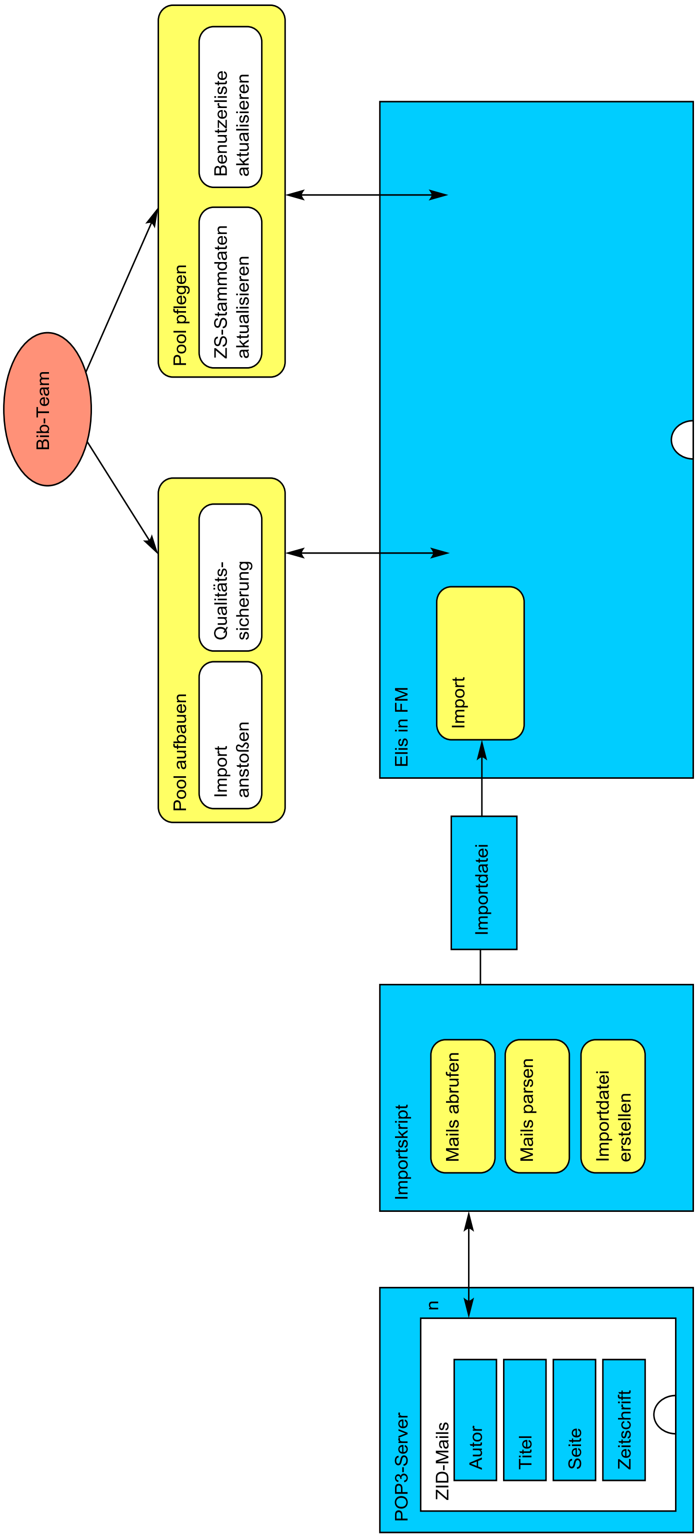


Figure 99: Fill Pool with ToC entries

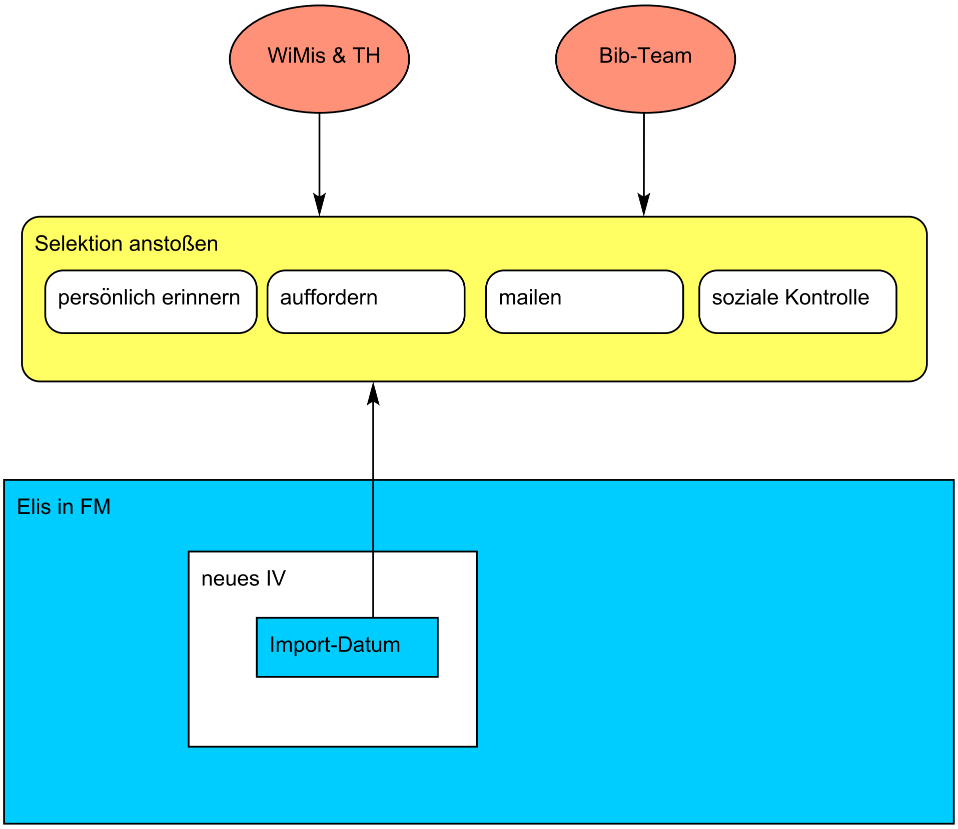


Figure 100: Trigger Selection

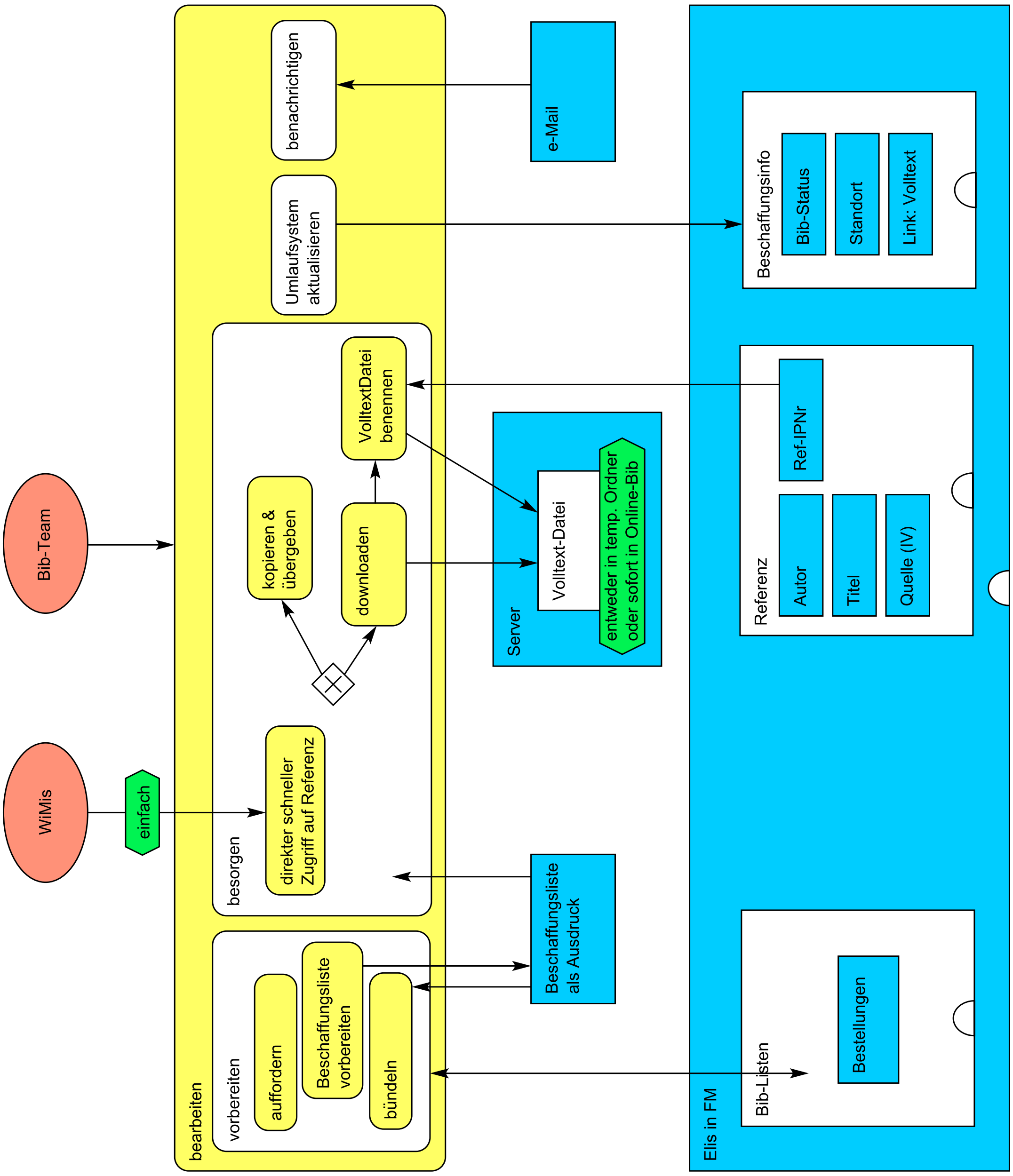


Figure 101: Get Articles

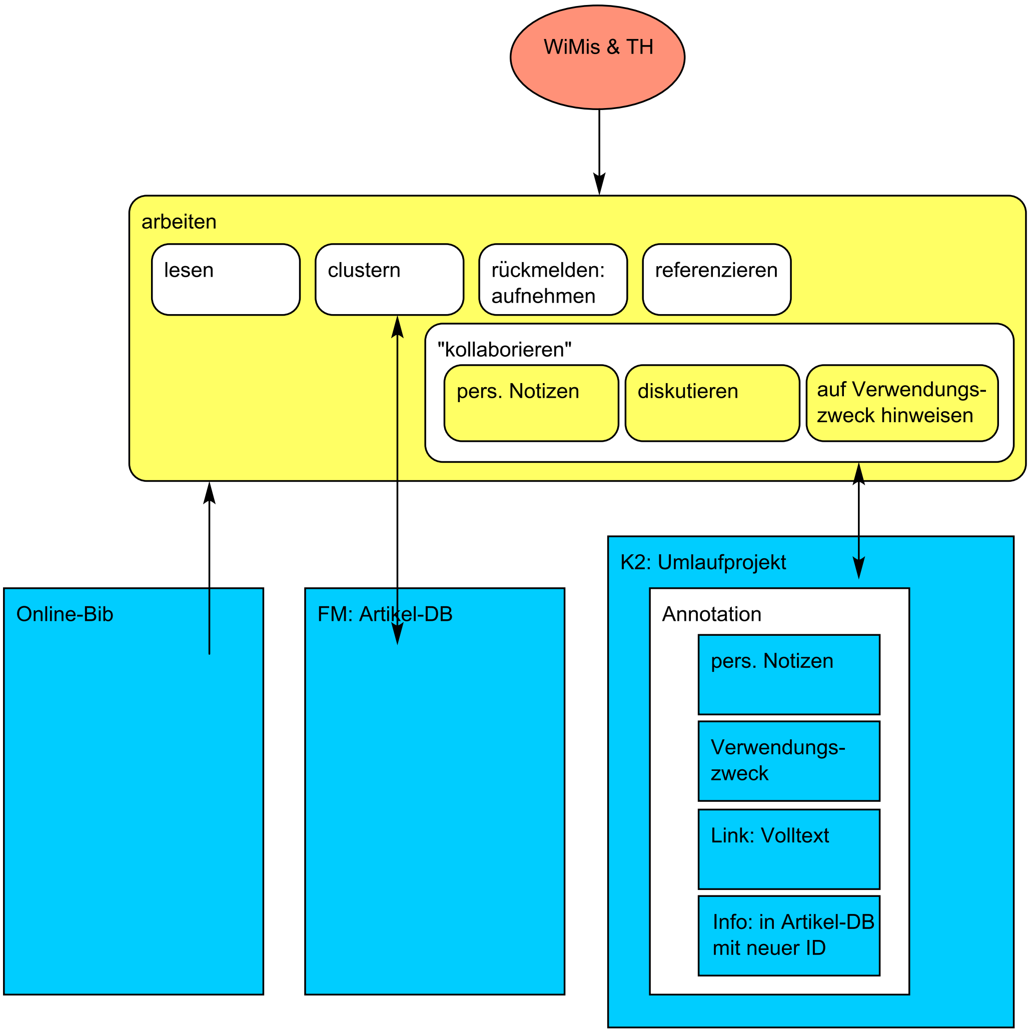


Figure 102: Access Literature during Research Work

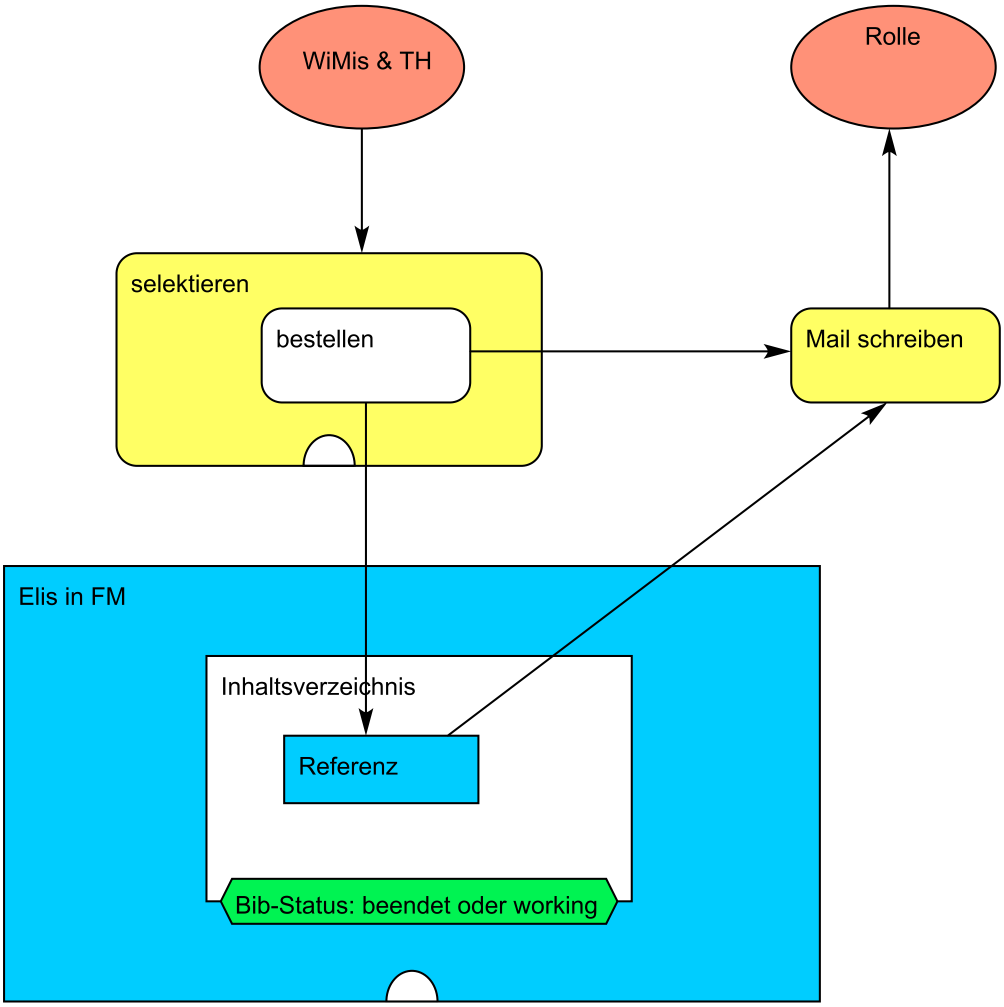


Figure 103: How to Order an Article when ToC has already been processed

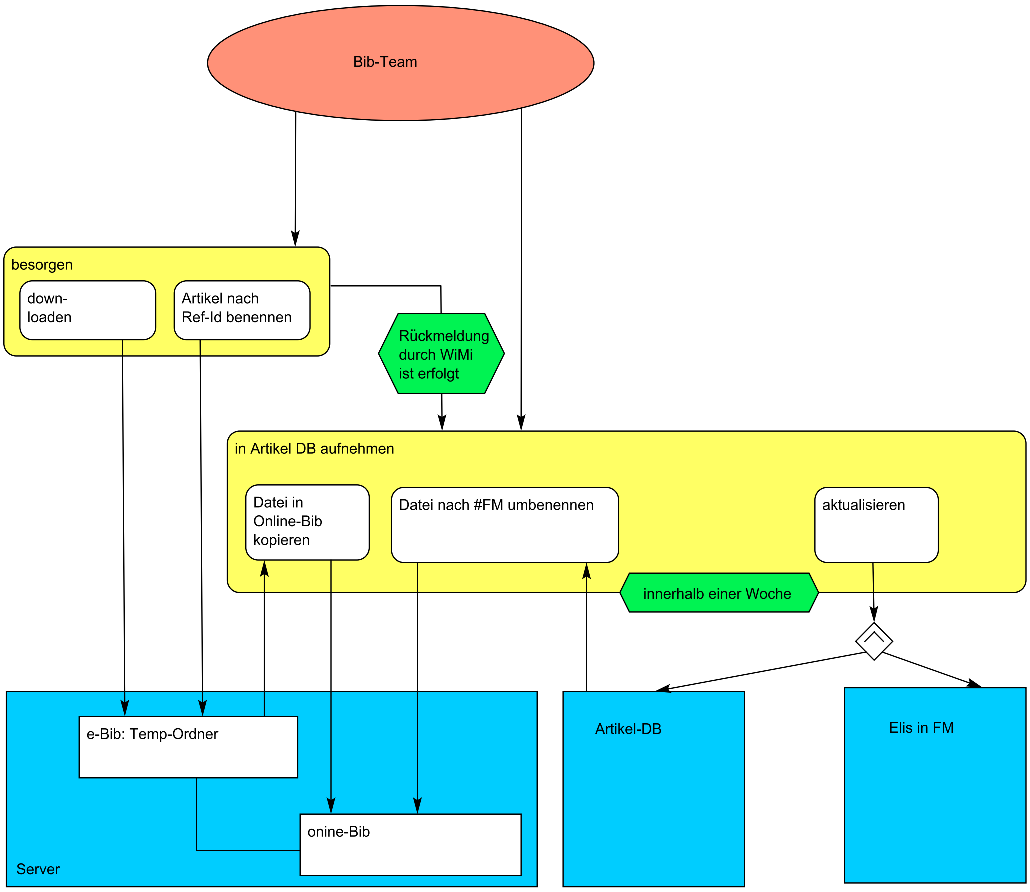


Figure 104: When to Include Articles into the Database?

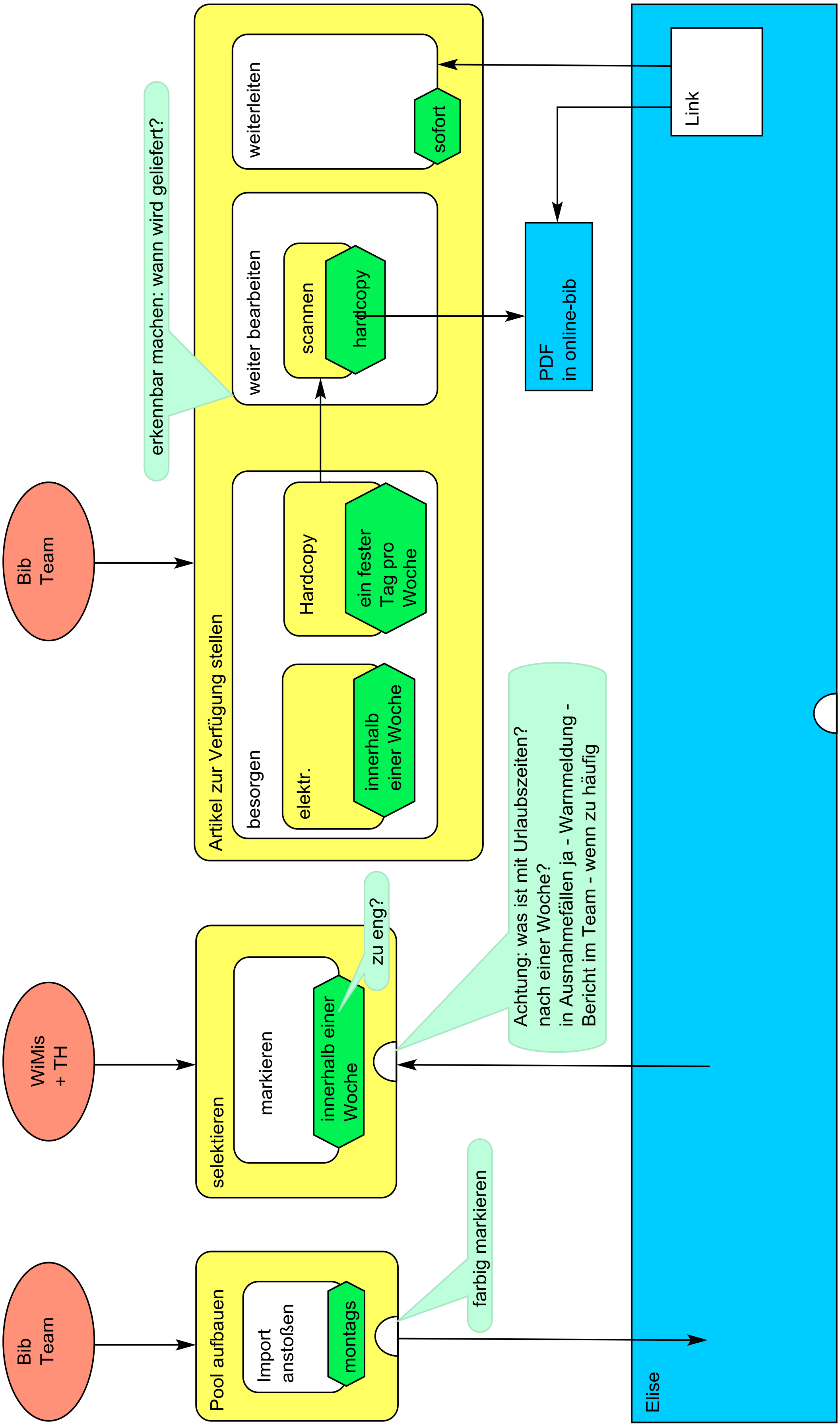


Figure 105: Delivery of Selected ToC Items

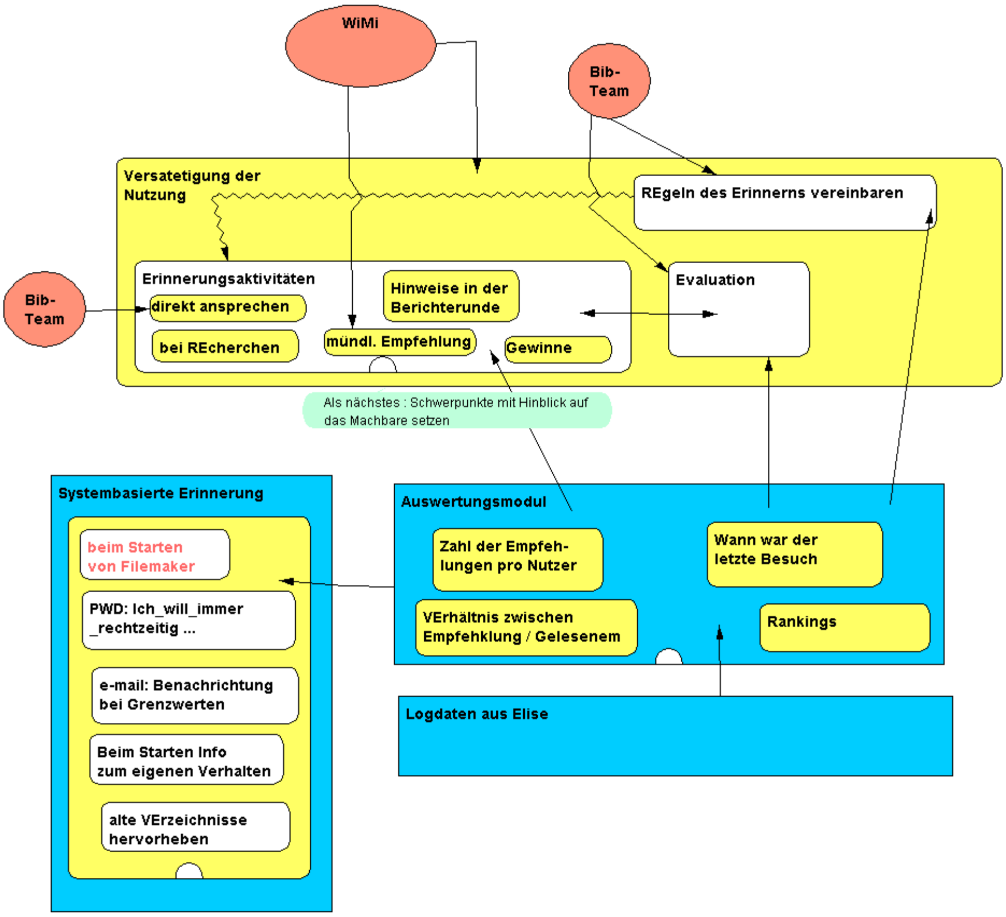


Figure 106: Ideas for Additional Awareness Features

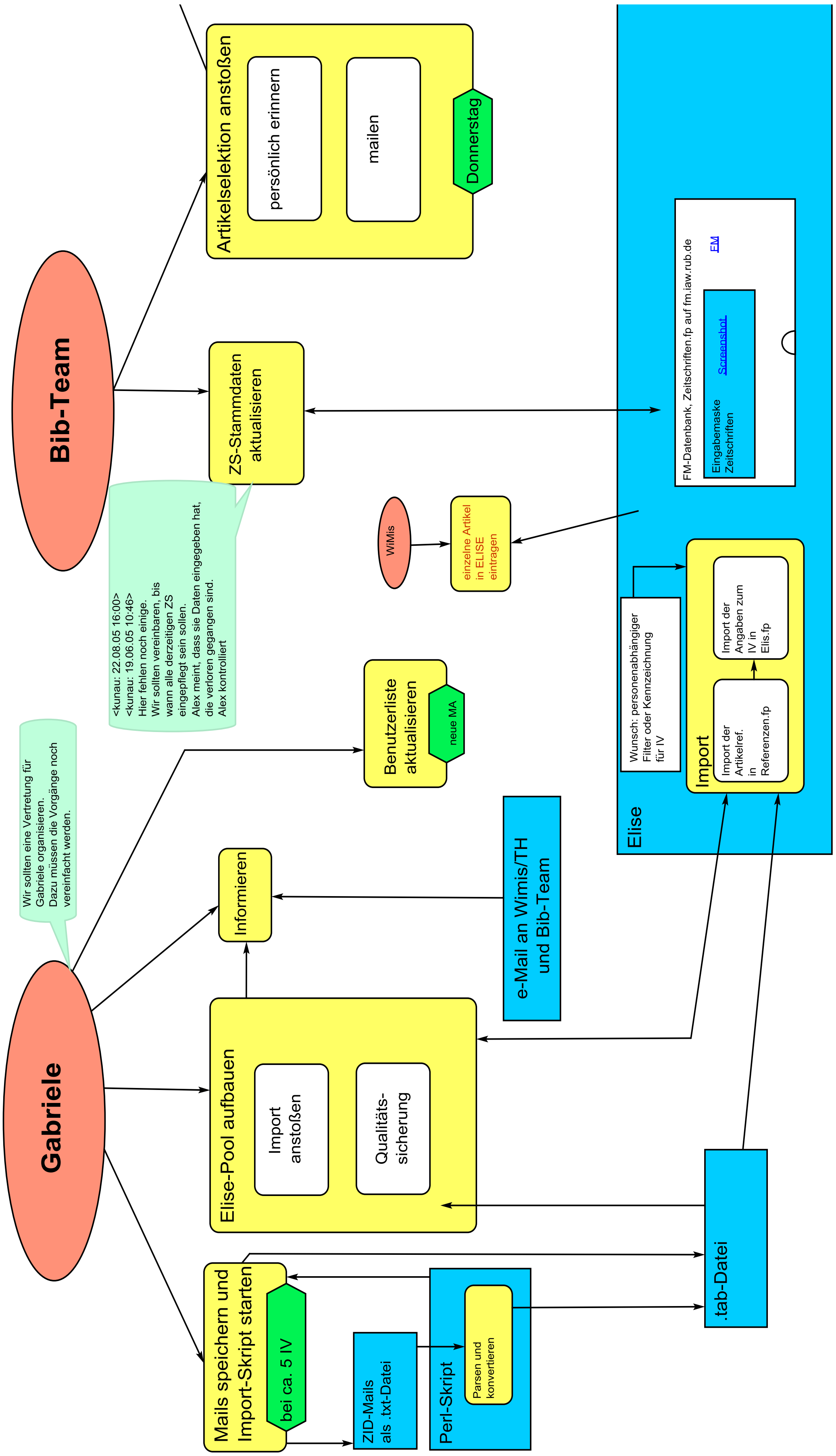
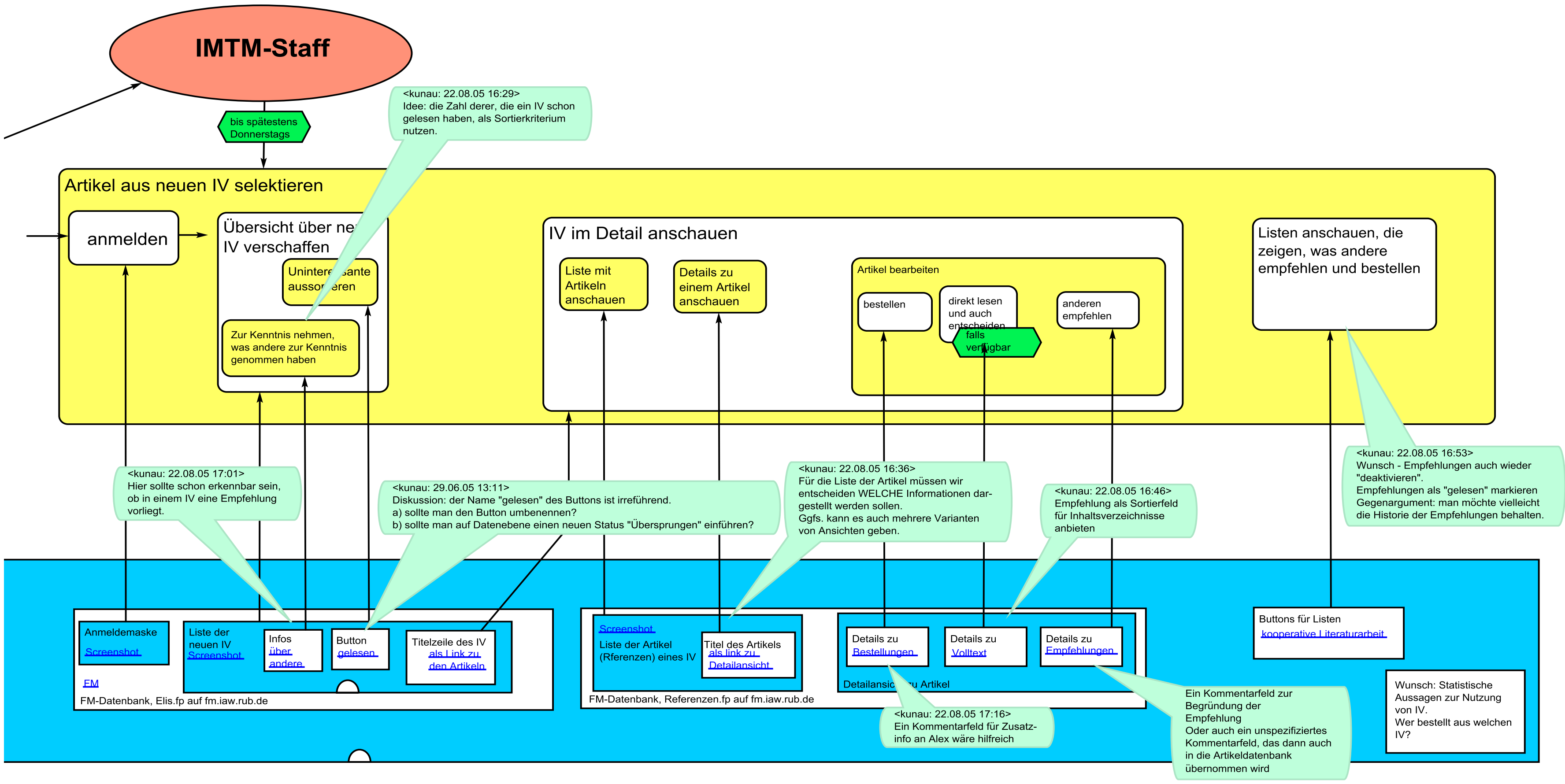

Bib-Team
Artikelselektion anstoßen
persönlich erinnern
mailen
Donnerstag
ZS-Stammdaten aktualisieren
<kunau: 22.08.05 16:00>
<kunau: 19.06.05 10:46>
Hier fehlen noch einige.
Wir sollten vereinbaren, bis wann alle derzeitigen ZS eingepflegt sein sollen.
Alex meint, dass sie Daten eingegeben hat, die verloren gegangen sind.
Alex kontrolliert
WiMis
einzelne Artikel in ELISE eintragen
Benutzerliste aktualisieren
neue MA
Wir sollten eine Vertretung für Gabriele organisieren.
Dazu müssen die Vorgänge noch vereinfacht werden.
Gabriele
Informieren
e-Mail an Wimis/TH und Bib-Team
Elise-Pool aufbauen
Import anstoßen
Qualitäts-sicherung
Mails speichern und Import-Skript starten
bei ca. 5 IV
ZID-Mails als .txt-Datei
Perl-Skript
Parsen und konvertieren
.tab-Datei
Elise
FM-Datenbank, Zeitschriften.fp auf fm.iaw.rub.de
EM
Eingabemaske Zeitschriften
Screenshot
Wunsch: personenabhängiger Filter oder Kennzeichnung für IV
Import
Import der Artikelref. in Referenzen.fp
Import der Angaben zum IV in Elis.fp

Figure 107: First Diagram after Evaluation Workshop (Part 1 of 2)


IMTM-Staff
bis spätestens Donnerstags
Artikel aus neuen IV selektieren
anmelden
Übersicht über ne IV verschaffen
Unintere sante ausso eren
Zur Kenntnis nehmen, was andere zur Kenntnis genommen haben
IV im Detail anschauen
Liste mit Artikeln anschauen
Details zu einem Artikel anschauen
Artikel bearbeiten
bestellen
direkt lesen und auch entscheiden
falls verfügbar
anderen empfehlen
Listen anschauen, die zeigen, was andere empfehlen und bestellen
Anmeldemaske Screenshot
Liste der neuen IV Screenshot
Infos über andere
Button gelesen
Titelzeile des IV als Link zu den Artikeln
FM
FM-Datenbank, Elis.fp auf fm.iaw.rub.de
Screenshot
Liste der Artikel (Rferenzen) eines IV
Titel des Artikels als link zu Detailansicht
FM-Datenbank, Referenzen.fp auf fm.iaw.rub.de
Details zu Bestellungen
Details zu Volltext
Details zu Empfehlungen
Detailansic u Artikel
Buttons für Listen kooperative Literaturarbeit

<kunau: 22.08.05 16:29>
Idee: die Zahl derer, die ein IV schon gelesen haben, als Sortierkriterium nutzen.

<kunau: 22.08.05 17:01>
Hier sollte schon erkennbar sein, ob in einem IV eine Empfehlung vorliegt.

<kunau: 29.06.05 13:11>
Diskussion: der Name "gelesen" des Buttons ist irreführend.
a) sollte man den Button umbenennen?
b) sollte man auf Datenebene einen neuen Status "Übersprungen" einführen?

<kunau: 22.08.05 16:36>
Für die Liste der Artikel müssen wir entscheiden WELCHE Informationen dargestellt werden sollen.
Ggfs. kann es auch mehrere Varianten von Ansichten geben.

<kunau: 22.08.05 16:46>
Empfehlung als Sortierfeld für Inhaltsverzeichnisse anbieten

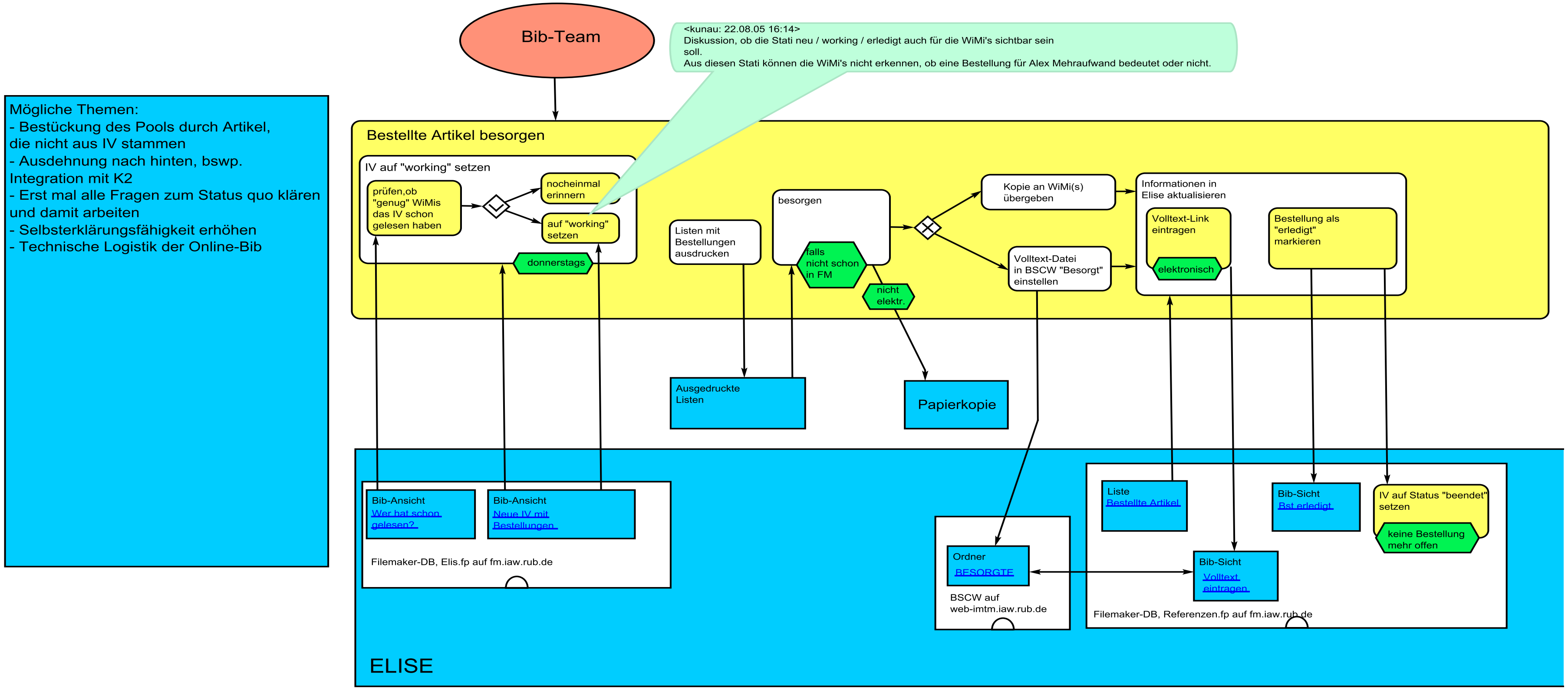
<kunau: 22.08.05 16:53>
Wunsch - Empfehlungen auch wieder "deaktivieren".
Empfehlungen als "gelesen" markieren
Gegenargument: man möchte vielleicht die Historie der Empfehlungen behalten.

<kunau: 22.08.05 17:16>
Ein Kommentarfeld für Zusatz-info an Alex wäre hilfreich

Ein Kommentarfeld zur Begründung der Empfehlung
Oder auch ein unspezifiziertes Kommentarfeld, das dann auch in die Artikeldatenbank übernommen wird

Wunsch: Statistische Aussagen zur Nutzung von IV.
Wer bestellt aus welchen IV?

Figure 108: First Diagram after Evaluation Workshop (Part 2 of 2)

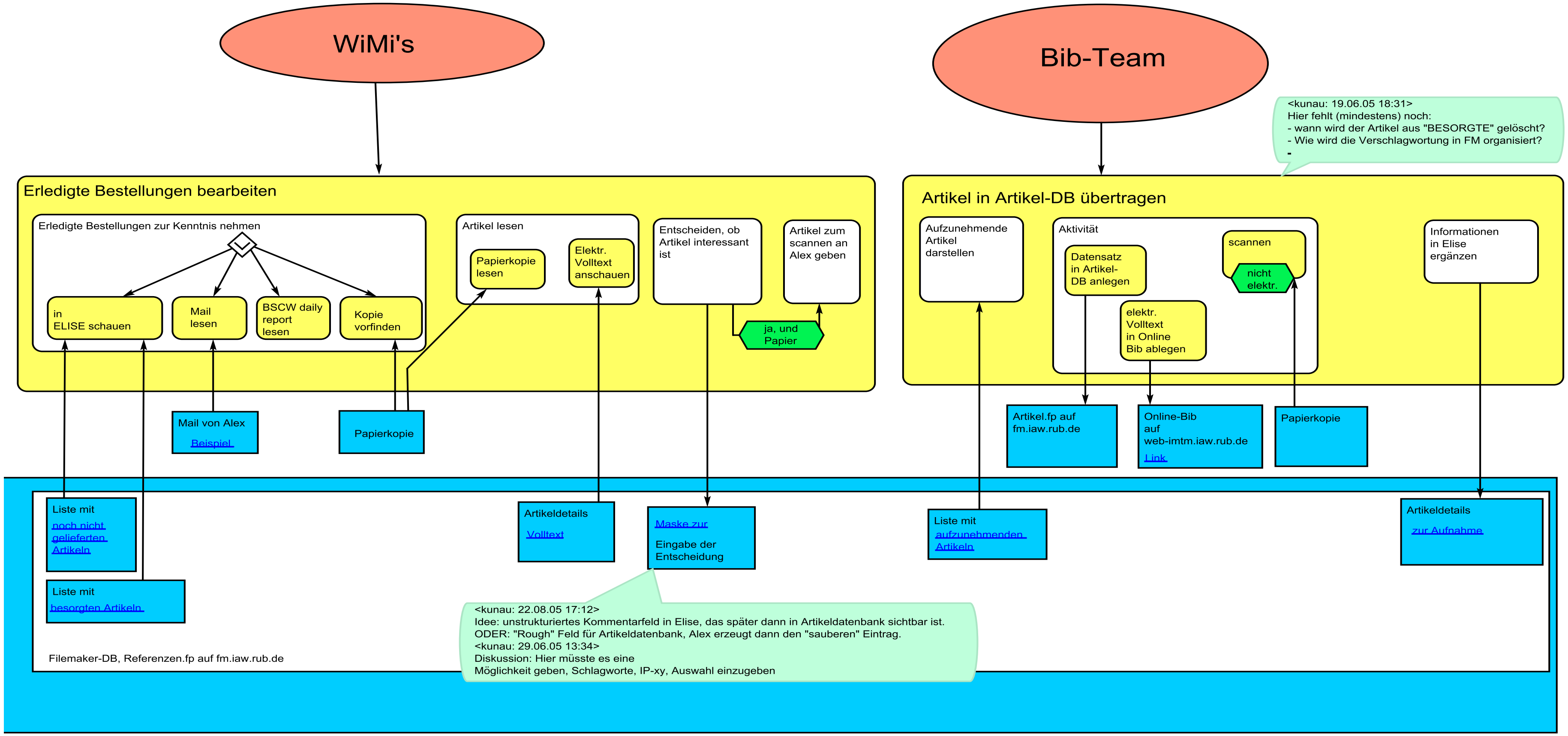


<kunau: 22.08.05 16:14>
Diskussion, ob die Stati neu / working / erledigt auch für die WiMi's sichtbar sein soll.
Aus diesen Stati können die WiMi's nicht erkennen, ob eine Bestellung für Alex Mehraufwand bedeutet oder nicht.

Mögliche Themen:
- Bestückung des Pools durch Artikel, die nicht aus IV stammen
- Ausdehnung nach hinten, bswp. Integration mit K2
- Erst mal alle Fragen zum Status quo klären und damit arbeiten
- Selbsterklärungsfähigkeit erhöhen
- Technische Logistik der Online-Bib

Bib-Team
Bestellte Artikel besorgen
IV auf "working" setzen
prüfen,ob "genug" WiMis das IV schon gelesen haben
nocheinmal erinnern
auf "working" setzen
donnerstags
Listen mit Bestellungen ausdrucken
besorgen
falls nicht schon in FM
nicht elektr.
Kopie an WiMi(s) übergeben
Volltext-Datei in BSCW "Besorgt" einstellen
Informationen in Elise aktualisieren
Volltext-Link eintragen
elektronisch
Bestellung als "erledigt" markieren
Ausgedruckte Listen
Papierkopie
Bib-Ansicht Wer hat schon gelesen?
Bib-Ansicht Neue IV mit Bestellungen
Filemaker-DB, Elis.fp auf fm.iaw.rub.de
Ordner BESORGTE
BSCW auf web-imtm.iaw.rub.de
Liste Bestellte Artikel
Bib-Sicht Bst erledigt
IV auf Status "beendet" setzen
keine Bestellung mehr offen
Bib-Sicht Volltext eintragen
Filemaker-DB, Referenzen.fp auf fm.iaw.rub.de
ELISE

Figure 109: Second Diagram after Evaluation Workshop (Part 1 of 2)


WiMi's
Bib-Team
Erledigte Bestellungen bearbeiten
Erledigte Bestellungen zur Kenntnis nehmen
in ELISE schauen
Mail lesen
BSCW daily report lesen
Kopie vorfinden
Artikel lesen
Papierkopie lesen
Elektr. Volltext anschauen
Entscheiden, ob Artikel interessant ist
ja, und Papier
Artikel zum scannen an Alex geben
Mail von Alex
Beispiel
Papierkopie
Artikel in Artikel-DB übertragen
Aufzunehmende Artikel darstellen
Aktivität
Datensatz in Artikel-DB anlegen
elektr. Volltext in Online Bib ablegen
scannen
nicht elektr.
Informationen in Elise ergänzen
Artikel.fp auf fm.iaw.rub.de
Online-Bib auf web-imtm.iaw.rub.de
Link
Papierkopie
Liste mit noch nicht gelieferten Artikeln
Liste mit besorgten Artikeln
Artikeldetails
Volltext
Maske zur Eingabe der Entscheidung
Liste mit aufzunehmenden Artikeln
Artikeldetails
zur Aufnahme
Filemaker-DB, Referenzen.fp auf fm.iaw.rub.de

<kunau: 19.06.05 18:31>
Hier fehlt (mindestens) noch:
- wann wird der Artikel aus "BESORGTE" gelöscht?
- Wie wird die Verschlagwortung in FM organisiert?
-

<kunau: 22.08.05 17:12>
Idee: unstrukturiertes Kommentarfeld in Elise, das später dann in Artikeldatenbank sichtbar ist.
ODER: "Rough" Feld für Artikeldatenbank, Alex erzeugt dann den "sauberen" Eintrag.
<kunau: 29.06.05 13:34>
Diskussion: Hier müsste es eine
Möglichkeit geben, Schlagworte, IP-xy, Auswahl einzugeben

Figure 110: Second Diagram after Evaluation Workshop (Part 2 of 2)

11.4 Appendix – Transcripts from Workshops in Case Study 1

Another source of material that was used for the analysis of the case studies consists of the audio- and video recordings from workshops. Section 6.1 (pp. 151) discusses the research methods and section 11.1 (pp. 299) provides an overview of all empirical work and material.

This appendix contains transcripts of those passages from recordings that are explicitly used in the argumentation. The complete set of recordings is available on video-cassettes and DVDs.

11.4.1 Participants add Contents during Discussion of Diagrams

The following transcript stems from case study 'Mobile Communication System', STWT-1 (cf. 11.1.2, pp. 299). It is the passage that gave rise to the inclusion of two additional tasks in the drivers' work-procedures. Refer to section 7.1.1.1 for a discussion.

(Source: Datei 07 - AUFN 2002 12 17 STWT1 TEILA.mp3)

Person
D=Driver
PL=Project leader
F=Facilitator
	Time on
	Tape
		Excerpt

F	0:57:59	Gibt es sonst noch irgendetwas ... Bei der Vorstellung, ich gebe jetzt dieses Plakat sozusagen so ab an die Software-Entwickler, wo Sie sagen würden: „halt, das muss aber noch korrigiert werden“
D (W)	0:58:12	Ja, ich hab da nochmal was. Das hört sich alles so schön an da oben. Fahrer macht die Aufgabe, die und die. Das ist alles so Schema 15, dann ist das fertig. Da ist ... zum Beispiel, fahre ich irgendwo hin, und dann komme ich da an, sag „Schönen guten Tag, xy für Firma K. Ich hab Eisen auf'm Auto.“ „Ja, sagt die Hausfrau dann, ja, mein Mann hat mir Bescheid gesagt.“ Ich sag, „schön“. „Wir möchten das gerne hinten im Hof an der Garage haben.“ *Längere Pause* 700kg Eisen oder so. Wer lädt das ab? *Pause* Dann macht der Fahrer das. Der lädt dann eben 700 Kilo Eisen mit der Hand ab und schmeißt die vor der Garage. So Sachen, darum geht das auch.
D (xy)		Wenn du das nicht machst, rufen die direkt den Geschäftsführer an. ... „der Fahrer war ein Flegel“
		Allgemeines, zustimmendes Gerede
D (W)	0:59: 17	Oder da wird ein Auftrag gemacht vom K.. „Fahr mal dahin“. Und dann fährt man dahin, und dann fährt man in einen Feldweg rein, mit einem 16m – Zug, wo man überhaupt nicht mehr rauskommt. Aber da fahre ich 800m gerade in einen Feldweg rein, und dann habe ich die Addresse gefunden. Ja und dann stehe ich da und weiß nicht, wie ich da wieder

Person
D=Driver
PL=Project leader
F=Facilitator
	Time on
	Tape
		Excerpt

		rauskommen soll. Weil ich da nicht drehen kann, weil ich da in die Matsche rein fahr. Aber das ist auch eine Sache, wo die Firma K.
D (B)		... das habe ich schon oft angeprangert, das gehört auf den ... *???* ... ... bei Kunden hier, die sind im reinen Wohngebiet, verkehrsberuhigt, und dann mit einem Sattel da rein ... sehr schön, wenn man dann 500m rückwärts wieder rumrücken muss, um da wieder rauszukommen, ist natürlich auch sehr schön.
D or PL		Aber da sind viele Sonderfälle, und da haben wir ...
F	1:00:26	Damit wir es nicht vergessen, also das Eine wäre „selber abladen, wenn keine Ablademöglichkeiten zur Verfügung stehen“ ... *Draftsman works on diagram*
D		Ja ... *zustimmendes Reden ...*
D		... eine Stunde vorher avisieren, damit du anrufst, dass die Leute auch weg sind.
		... *zustimmendes Lachen ...*
D		„Lass den Bekloppten mal machen“
		...
F		*Speaks with assistant who handles the editor to draw the diagrams* Zeichne doch mal „Gelände wieder verlassen“ wenn das so Problem ist, wenn man nur rein kommt und nicht wieder raus.
D	1:01:43	Na ja, „nicht wieder raus“ ... raus musst du ja sowieso, irgendwie. Geht ja nun mal nicht anders. Aber stehst auf einmal im Dreck, oder im Feldweg, oder beim Bauer ...
		... *weitere zustimmende Aussagen dazwischen ...*
D (B)		Es muss direct angegeben sein, hier mit kleinem Fahrzeug anliefern
		... dabei *weiteres Reden mit noch mehr Beispielen ...*

11.4.2 Postponing of Organizational Decisions

The following passage from STWT-2 is an example where software engineers need less information for their work than is necessary to plan computer-supported work-processes. Topic of the discussion was they way in which the dispatcher accesses the tour-data available with the new system. Should the data be automatically transferred and even displayed in a pop-up window, or should there be a mere pull-mechanism, by which the dispatcher can request information when needed? For a discussion refer to Example 44, p. 240).

(Source: Case-Study 1, STWT-2, Aufnahme „Teil B”)

Person **D=Driver** **Dp=Dispatcher** **PL=Projectleader** **F=Facilitator** **SW=Software eng.**	**Time on Tape**	**Excerpt**
F	13:40	Wir haben die sortierte Route im System. Sie sagen, jetzt interessiert sich der Disponent dafür. Das heißt, wir bräuchten jetzt eine Möglichkeit, dass der Disponent ...
PL		... das abgreift ...
F		... die ... die abgreift
PL		Und zwar nicht nur einmal, also ich meine, der Fahrer ... zum Beispiel während der Tour umplanen muss ... ist doch dasselbe ... ... es ist für die Disposition doch äußerst sinnvoll, wenn ich weiß, er wird da und da in dem und dem Ort leer, dann kann ich bspw. eine Abholung noch planen, oder, oder, oder,
		...
F	14:15	... Abfrage durch den Disponenten
D?		Die aktuellen Tourdaten, ja sicher ...
PL		... entweder fragt der ab, oder es wird immer wenn eine Änderung gemacht wird automatisch gesendet. Ich weiß nicht wie man's macht, also
F	14:35	Wie würden Sie das sehen? Sollten Änderungen immer automatisch kommen, oder
Dp (M)		Nach Bedarf, weil immer ist es ja nicht wichtig, sag ich mal. Eigentlich ist ja nur in dem Moment wichtig, wenn ich Nachfragen habe, oder wenn mal wieder so ein Verkäufer nachfragt.
PL		Richtig, dann würdest du in dem Fall abfragen, oder du würdest reingucken und hättest schon immer den letzt-gesendeten Stand. Das ist die Frage, ob du abfragst – ich meine, ob du zum Beispiel sagst, welcher Wagen wäre heute Mittag am nächsten bei dem und dem Kunden zum Abholen. Dann wäre der letzte Stand eigentlich am Besten, oder nicht? Also seh' ich so. Aber ich weiß nicht, was da darstellbar ist.
SW	15:11	Das Problem mit dem Abfragen und so weiter ist schon mehrfach diskutiert worden. Es soll Möglichkeiten geben, dass die Daten zwischengespeichert werden und je nach Priorisierung gibt es Daten, die sofort aktualisiert wird und sofort kriegt der Disponent also irgendwie eine Meldung, dass
PL		... um Kosten zu sparen
SW		… da ist was Neues, also es sind verschiedene Sachen. Einerseits die Informationsflut – es wurde gesagt, dass der Disponent nicht permanent informiert wird „ich habe jetzt eine Kurve gefahren“ überspitzt gesagt, sondern, das interessiert den nicht, sondern wir wollen die Möglichkeit vorsehen, dass bestimmte Daten priorisiert werden. Also wenn DAS passiert, dann kriege ich immer und sofort eine Nachricht als Disponent, andere Sachen werden zwischen gehalten, die muss ich mir holen. Das läuft unter dem Stichwort push-und-pull Informationen. Also einige Informationen werden immer an den Disponenten geliefert, wie auch immer. Andere werden vorgehalten, die muss der

Person **D=Driver** **Dp=Dispatcher** **PL=Projectleader** **F=Facilitator** **SW=Software eng.**	**Time on** **Tape**	**Excerpt**
		Disponent sich holen. Und das wird priorisiert, und dann muss man irgendwie ... die Daten sind da und dann muss irgendwann ein Entscheidungsträger uns mal sagen, die, die, die und die mach' so, und die und die und die mach so, und dann wird man das einmal laufen lassen das System und wird feststellen, da fehlen noch ein paar Informationen und paar anndere müssen umstrukturiert werden, das ist normal. Aber dieses Verfahren ist konzeptionell vorgesehen
Dp (M)	16:19	... aber …
PL		… was …
SW		Aus Kostengründen, aus Informationsflutgründen, alles mögliche ... ja bitte ...
Dp (D)		Wer profitiert jetzt von dem System ...
		… Diskussion darüber, wer davon profitiert, dass Informationen nun schneller vorliegen.
F	23:00	Ich würde gerne einen Punkt noch einmal aufnehmen. Wir hatten gerade zwei Fragen. Soll eine Änderung der Tour automatisch auf Ihrem Bildschirm auftauchen, oder sollte es eher so sein, dass Sie, wenn Sie sich für die Route eines Fahrer interessieren, dass Sie nachgucken. Da hatte ich jetzt verstanden, dass das Automatische, das Aufpoppen, nicht gewünscht ist.
D (M)		Ne, ich würde sagen nicht
D ?		Ne
D (M)		Nachgucken
F		Nachgucken
PL		Das reicht doch auch

11.4.3 Technical Alternatives – Use of Mobiles or SpiW-Com

This transcript of STWT-2 documents the first time that the topic, whether or not mobiles should be used in parallel with the new mobile communication system came up (cf. Example 17, p. 195).

(Source: STWT-2, Aufnahme Teil B)

Person **D=Driver** **Dp=Dispatcher** **PL=Projectleader** **F=Facilitator** **SW=Software eng.**	**Time on** **Tape**	**Excerpt**
Dp (D)	34:18	Ich mein, dass man, wenn das System mal läuft, und man kennt sich dann damit aus, dass ich in brandeiligen Fällen, ihn trotzdem nicht anrufe, und sag ... ist ein bisschen weit ... oder

Person
D=Driver
Dp=Dispatcher
PL=Projectleader
F=Facilitator
SW=Software eng.
	Time on Tape	Excerpt

		wie auch immer
SW		Ich weiß nicht, wie wir das einführen Herr D. Die Handys werden doch alle eingezogen, dann, oder nicht?!
		Lachen
PL (?)		Nein
SW.		Natürlich nicht. Ganz ernst, das ist ein unterstützendes System ...
D (B)	34:40	Naja, dann ... wir halt durch Trommeln; mein Gott, was soll's, ja?
SW		Nein, nein, das ist ein Unterstützungssystem und ... das man nutzen kann, um eben, wie gesagt, so händische Information ... immer dieses permanente Anrufen ... bestimmte Information kriegt man automatisch ... kann die automatisch verfügbar machen ... kann die automatisches Abgreifen, dann hab' ich es einfach einfacher.
Dp		Ok

11.4.4 SpiW-Com to Check on the Drivers

This transcript from STWT-2 illustrates how a software engineer implies that his interpretation of the CSCW-system will be shared by the organization. The topic discussed is whether and how the new data (e.g. time of departure; actual position …) available to the dispatcher will be used to check on the drivers (cf. Example 43, p. 239).

(Source: Case Study 1, STWT-2, Aufahme Teil B)

Person
D=Driver
Dp=Dispatcher
PL=Projectleader
F=Facilitator
SW=Software eng.
	Time on Tape	Excerpt

PL	36:18	Also, ich find, die Fahrtstrecke zu kontrollieren, finde ich vermessen. Weil, das ist wirklich Sache des Fahrers meiner Meinung nach ... ich mein, ich werd' mich sicher fragen, wenn ich merke, er hat für eine Strecke von 30 km 500 km zurück gelegt, aber ... im Endeffekt bestimmst Du, auf welcher Straße du schneller bist. Deswegen würde ich die nicht kontrollieren.
D	36:33	... Wenn der jetzt hier guckt, „aha, der ist jetzt beim B. gewesen, der nächste ist jetzt K.," und die Hauptkreuzung ist gesperrt, und

Person **D=Driver** **Dp=Dispatcher** **PL=Projectleader** **F=Facilitator** **SW=Software eng.**	**Time on** **Tape**	**Excerpt**
		die schicken mich jetzt hier erstmal Richtung Hennesee über Remmlinghausen, ja, und dann hinten über Westernmöhlefeld dann hinten wieder runter, ja, dann drückt der auf den Knopf und guckt „wieso, was macht der denn jetzt da oben", der will doch jetzt zum K.
Dp (D)		Das wird bestimmt nicht der Fall sein.
SW	36:55	Also, das ist nicht der Fall! Da kann ich Sie auch beruhigen. Das kann nicht der Fall sein. Das System kann ja nur dazu benutzt werden, mal nachzufragen. Also es ist ja kein Kontrollsystem, sondern die Daten werden erhoben, und die guckt man sich, irgendjemand guckt sich die mal an und dann fragt man mal nach ... und dann sagt derjenige, ich bin mal ne Umleitung gefahren, da war gesperrt. So. Dieses System kommt ja erst dann zum Tragen, wenn das wiederholt vorkommt ... da passieren immer wieder komische Sachen irgendwie in Anführungsstrichen „komische Sachen" und dafür ist das gedacht und nicht als Kontrollsystem wie Herr D. schon sagt, da fährt er rechts rum statt links rum und ich finde doch links ist doch die viel schönere Strecke und rechts sind viel mehr Bäume, oder was auch immer ... Das ist nicht damit gedacht. Die Daten sollen einfach erhoben werden, um einerseits den Auftragsfortschritt zu kontrollieren, und andererseits um Erkenntnisse für die Planung und auch für das Zukünftige daraus abzuleiten. Nicht um zu gucken, welche Straße Sie da jetzt nun fahren.
F		So dass wir jetzt also irgendein Feld für eine Eingabe, eine Begründung, warum Sie einen gewissen Weg ausgewählt haben, das würden wir nicht ...
PL		Also ich find es jetzt auch nicht notwendig, dass der Disponent sieht, ja, eine Karte, wo jetzt dein roter Fahrweg dargestellt ist. Sondern für mich würde als Disponent wirklich reichen, wenn sag ich mal, l Stopps, wo du gestanden hast, und dass du gerade fährst. Die Information die reicht mir.
		...
F	38:20	Also eine Begründung für die ausgewählte Route geben wir nicht ins System ein.
		Nein

11.5 Appendix – Transcripts from Interviews / Questionnaires

11.5.1 „Mobile Communication" – STWT-4

A few days after the qualification workshop (STWT-4) took place the participants were interviewed. The main purpose of the interviews was to provide material for a second research project concerning the knowledge-integration of the participants. But some questions were added for evaluating the usage of the diagrams; these are documented in the next sections.

11.5.1.1 Question 6

Question

Inwiefern finden Sie jetzt, nach dem Workshop, notwendig und sinnvoll, dass die Disponenten und die Fahrer Vereinbarungen für ihre Kooperation mit der Nutzung von der neuen Software aufstellen? Warum?

Answer – Project Leader / Dispatcher

Das halte ich für sehr notwendig, weil sonst das System von jedem anders verstanden und genutzt wird. Daraus resultiert, dass die die Nachfragen per Handy kommen müssen, sonst versteht der eine gar nicht, was der andere ihm schickt. Das heißt, es muss klar sein, wenn ich die Information kriege „Hier ist das Material rostig" – ist das rostig oder nicht richtig rostig, sondern leicht gerostet oder, oder, oder. Ich denke, dass ohne das Vereinbaren von irgendwelchen Regeln auf Dauer gar nicht umzusetzen ist.

Answer – Manager

Ich finde es schon sinnvoll, weil wir einfach darüber von vornherein irgendwelche Streitpunkte aus dem Weg geräumt haben. Weil man Klarheit in die Prozesse rein bringt, die vorher vielleicht nicht ganz klar sind.

Answer – Software engineer

Auf jeden Fall notwendig... Also, es ist herausgekommen im Workshop, dass das System nicht implizit alles abdeckt, sondern dass explizite Vereinbarungen erforderlich sind. Wie zum Beispiel, wann ist anzurufen. Das hat die Tatsache, die Autonomie des Fahrers zu erhöhen, er kann entscheiden, wann er das tut. Aber für diesen Entscheidungsspielraum, da sind natürlich bestimmte Handlungsanweisungen notwendig oder erforderlich.

Answer – Driver 1

Ja, ich würde es schon wichtig sehen... ja, warum? Wichtig ist das Ganze drum und dran, was wir da hatten, mit Einweisung und so, das war gute Sache und das war, ich glaube, auch von den anderen eine gute Lösung für dieses technische gebacken.

Answer – Driver 2

Das ist sehr wichtig, sonst funktioniert das ganze System nicht.

11.5.1.2 Question 7

Question

7. Wie hilfreich haben Sie die Modellierung im Workshop empfunden?

- für die Diskussion über die Vereinbarungen?

- für die Dokumentation?

7a. Hat Sie etwas irritiert?

Woran lag es?

- an der Projektion?

- An der Darstellungssymbolen?

- An der Modellierung überhaupt?

- An der Aufgabe?

7b. Was würde Ihnen helfen?

Answer – Project Leader / Dispatcher

HD: In Gesprächen an sich ist die Art der Modellierung für mich eher verwirrend gewesen. Weil in dem Gespräch, wenn da ein Kästchen aufgemacht wurde, wenn da was geschrieben wurde, wenn ein Pfeil irgendwohin gezogen wurde, da hätte ich gesagt: „Okay, pass mal auf, da Flipshart, da ist ein Punkt, der muss rein“. Weil ich in dem Gespräch in dem Thema drin bin, hätte ich da eher der Sache gefolgt. Also, ich bin im Gespräch, in dem Einzelnen, nicht nachgegangen, warum kommt jetzt ein Pfeil da und dahin, sondern ich habe lediglich meine Anmerkung gebracht und habe gesehen, die ist irgendwo rein geschrieben worden, und habe dann eigentlich nicht hinterfragt, hier warum muss das jetzt so sein, warum ist das ein Extra-Punkt oder, oder, oder. Wenn ich mir nach so einem Workshop diese Modellierung allerdings angucke, ist sie für mich, von der Systematik wesentlich eher zu verstehen, ich glaube auch als Außenstehender, als es für einen wäre, der das Ganze jetzt in Stichpunkten mit so einer Handout entsprechend bekommen würde ohne diese Modellierung.

IL: Zum Beispiel das hier (zeigt den Ausschnitt aus dem Protokoll mit Meta-Plan Punkten)

HD: Richtig, genau. Für die spätere Nachvollziehbarkeit ist die Modellierung sehr gut eigentlich. Sie in den Gespräch wirklich reinzubringen, muss ich ganz ehrlich sagen, weiß ich nicht, ob das… Da habe in den einzelnen Diskussionen wenig damit gearbeitet.

IL: Würden Sie sagen, für die Diskussion an sich war das eher ablenkend oder nicht so förder-lich… Wir unterstützen auch die Entstehung neuer Ideen und dass dann die Leute Bezug auf-einander nehmen und versuchen damit, Diskussion zu fördern…

HD: Ich kann mir vorstellen, dass es sogar verwirrend ist.

IL: Also, für die Diskussion verwirrend, als Dokumentation gut, im Nachhinein.

HD: Ja

IL: Was würden Sie uns raten? Dies nicht zu tun, es anders tun, die Modellierung während der Diskussion. Oder sollten wir die Leute anders vorbereiten?

HD: Was würde ich raten… Ich glaube, ich würde es anders tun. Ich würde das Visualisieren und auch Festhalten von Ideen, die von Workshopteilnehmern kommen, finde ich grundsätzlich gut. Denn nur daraus resultieren oder kommen andere Ideen, von den Workshopteilnehmern. Finde ich grundsätzlich in Ordnung. Die Frage ist, ob ich die Punkte einfach sammele, unter gewissen Überschriften, sage okay, was passiert alles da und dann nachhinein sortiere, vielleicht auch auf dem Laptop etwas modelliere, aber das den Workshopteilnehmern nicht zeige, sondern sage, okay, ich mache mein Flipshart, mache gewisse Überschriften, sage, worauf müssen wir achten beim Empfänger und dann wird es in den Raum geworfen. Dann steht es da, der Fahrer oder Disponent kann gucken, okay, den Punkt hätte ich gerne noch dabei. Noch mal, das Nachgehen von Pfeilen ist natürlich, ist natürlich für Sie in der Diskussion schon wichtig, weil Sie festlegen, welche Abläufe möchte ich in welcher Reihenfolge haben, aber ich glaube, dass der Workshopteilnehmer an sich dadurch etwas gehemmt ist.

Answer – Manager

HB: Sie meinen das, was an diesen Tafeln gemacht wurde?

IL: Also, ich meine nicht diese Tafeln [zeigt Foto mit Meta-Plan], das bezeichne ich nicht mit Modellierung. Das sind Modelle [zeigt Modelle im Protokoll].

HB: Also, das hier.

IL: Ja. FK hat das versucht, was gesagt wurde, zu visualisieren, in Modell einzubauen, in die Arbeitschritte des Fahrers. Und das hat sie während der Diskussion gemacht. Haben Sie auch verfolgt.

HB: Ja, ja. Das hat sie abgeändert hier und es ist sofort eingesetzt und aufgenommen worden.

IL: Ja. Fanden Sie das hilfreich?

HB: Ja, das war natürlich eine hilfreiche Unterstützung. Dass man es direkt eingesetzt kriegte, dann hatte man das vor Augen. Dann war es natürlich eine schöne Sache.

7a. Hat Sie etwas irritiert?

HB: irritiert? Kann ich nicht sagen.

IL: Weil wir hatten den Eindruck, dass die Modelle ein bisschen störend waren für die Diskussion.

HB: Ich habe nicht so einen Eindruck. Das Modell war da und dass natürlich viel zu dem Modell geguckt würde, das war auch klar. Sollte wahrscheinlich Sinn der Sache sein. Dass man im Gespräch mit dem Modell, das auch aufnimmt und das ein bisschen vertieft.

Answer – Software engineer

HS: Ein Bild sagt mehr als 1000 Worte. Daher ist die Modellierung hilfreich. Welche Notation man dabei verwendet, ist Geschmacksache. Jede Methode hat ihre Berechtigung, ihre Vorteile und Nachteile. Berechtigung für SeeMe: Schwachstrukturierte Arbeitsprozesse können abgebildet werden und Nachteile: wenn weich modelliert wird, dann höre ich auf. Ist für mich Schluss mit dem Modellieren.

HS: Also, der Vorteil ist, dass durch die Notation schwachstrukturierte Arbeitsprozesse abgebildet werden können, das ist ganz klar. Das indiziert auch den Nachteil. Das heißt, bestimmte Analysen, die aufgrund eines Formalismus durchgeführt werden können, können natürlich nicht durchgeführt werden. Das liegt aber an der Natur der Dinge der Notationen.

7a. Hat Sie etwas irritiert?

HS: nichts. Kein Bruch erlebt.

Answer – Driver 1

HK: Also, dass wir es mitbekommen haben, Schritt für Schritt?

IL: Ja.

HK: Für das System sehr hilfreich, macht es aber im Endeffekt relativ kompliziert.

IL: Ja. Fanden Sie das hilfreich für die Diskussion der Vereinbarungen? Das war für die Diskussion hilfreich. Das war anregend. „Ich habe es gesehen und bin auf weitere Ideen gekommen“

HK: Richtig.

IL: Und als Dokumentation? Wie hilfreich war die entstandene Dokumentation für Sie?

HK: Die hat mir ganz gut gefallen, war auch hilfreich gewesen für mich, um mehr, noch mehr Vorstellungen von diesem System zu haben und noch vertrauter damit zu werden. Weil es dadurch schwieriger wurde, weil es immer mehr wurde.

IL: Mg. Sie würden sagen, es war gut, aber für mich zu viel.

HK: Momentan zu viel, zu schnell einfach, zu kurz dafür. Die Zeit war zu kurz.

IL: Meinen Sie den Workshop oder die Fahrerschulung, die vorher war?

HK: Die Fahrerschulung war zu kurz. Auf jeden Fall.

IL: Die Fahrerschulung war für uns eine Geschichte und der Workshop der weitere Schritt. Ich möchte mit Ihnen zunächst über den Workshop sprechen und dann zum Schluss sprechen wir über die Fahrerschulung.

HK: Ja

IL: Ich notiere es aber, dass sie zu schnell und zu kurz war.

HK: noch mehr Übungen, mehr Wiederholungen. Wenn man es ein Mal macht, dann sitzt man noch da drin nicht.

(7a. Hat Sie etwas irritiert?)

IL: Bleiben wir bei der Modellierung, hat Sie etwas irritiert?

HK: Ja.

IL: Es wäre interessant für uns, was.

HK: Dass wir probiert haben, in dem Workshop umzugehen, und dass man Schritte im Gerät weitergemacht hat, schnell in den nächsten Schritt übergegangen ist, ohne dass man jetzt ge-merkt hat, dass es etwas passiert ist, dass man etwas rübergeschickt hat, zu dem Disponenten oder dass man vielleicht nichts rüber geschickt hat.

IL: Das hat Sie irritiert.

HK: Das hat mich irritiert, als wir es ausprobiert haben.

IL: Ja, gut. Wir wissen, dass das Gerät noch nicht fertig ist. Das war wichtig, im Projekt noch mal zu sehen, dieses Gerät kann noch so nicht eingesetzt werden. Das war für uns die Lehre. Es interessiert mich aber im Interview nicht. Wir bleiben jetzt bei Modellen. Wir haben die Modellierung verwendet, um die Diskussion zu dokumentieren, um die Diskussion zu strukturieren. Wir wissen es aber nicht, ob es für Sie so okay war, wie beispielsweise für uns. Und deswegen: Hat Sie irgendwas gestört, haben Sie gedacht: „Was kommt denn jetzt?" oder „Ich hätte jetzt was anderes gemacht, andere Visualisierung gebraucht und nicht dieses Modell?".

HK: Das Modell war insgesamt ganz gut.

IL: Sie würden nicht sagen, mich hat es irritiert, dass FK vorne modelliert hat und das war irgendwie für das Gespräch störend.

HK: Ne, da konnte man nachvollziehen, was gerade besprochen wurde. Das war okay. Sehr gut sogar, diese Art und Weise.

11.5.1.3 Question 9

Question

9. Für die Fahrer:

9a. Wie hilfreich haben Sie die Modelle in der Fahrer-Schulung empfunden?

9b. Falls die Schulung der Fahrer im Westkreis durchgeführt werden soll, was sollte zusätzlich berücksichtigt werden? Was sollte verbessert bzw. anders gemacht werden?

Answer – Driver 1

IL: Wir haben auch in der Fahrer-Schulung Modelle verwendet. Ich habe an Modellen zu-nächst gezeigt, welche Schritte macht der Fahrer mit diesem Gerät. Fanden Sie diese Modelle hilfreich oder würden Sie sagen: „Das war nicht notwendig".

HB: Doch, das ist notwendig. Wenn man keine Ahnung hat, dann muss einem so was gezeigt werden, damit man das auch Schritt für Schritt machen kann. Wenn man das sich über so ein Modell angucken kann, ist natürlich…

IL: Weil wir haben es uns überlegt, müssen wir es so machen? Vielleicht reichen die Masken aus, wo MH sagt, jetzt kommt die Maske, jetzt kommt die Maske. Und diese Modelle sagen, hier zu diesen Schritten, da kommt die Maske. Da waren wir uns nicht sicher.

HB: Diese Modelle, die waren schon besser. Ich gehe davon aus, dass das auch für die nächste Schulung für die Fahrer in diesem Rahmen ablaufen sollte.

Answer – Driver 2

HK: Ja, klar, man muss ja wissen, mit welchen Gedanken, oder was man jetzt macht, wenn man den nächsten Schritt im Softwareprogramm macht.

IL: Also, wenn ich jetzt mit meinen Modellen gar nicht da wäre, und nur MH mit seinem Gerät, mit seiner Projektion, würde ihnen was fehlen… Sie würden sagen, das reicht nicht aus, das Modell ist schon wichtig.

HK: Ja, um es komplett zu verstehen. Sonst fehlt dieses. Es sei denn, sie sind im System schon enthalten, diese Hinweisschritte.

IL: Ist nicht.

HK: Dann ist es nötig, auf jeden Fall.

11.5.2 Electronic ToC-Service

After four workshops in April 2005 the participants were asked questions about the usefulness of the diagrams. All questionnaires contained 1 closed (Table 35) and 3 open questions (Table 36 – Table 38); the questionnaire for the last workshop included one additional closed question (Table 39). The questionnaires were sent out by e-mail a few hours after the workshop; the participants answered within a few days by e-mail as well. Since the number of participants is so small, no quantitative analysis is made; the answers to the questionnaires are merely described in the following tables.

		4-Apr	12-Apr	19-Apr	26-Apr
How do you judge the usefulness of the contents in the diagrams of today's meeting?	They helped much			2	2
	They helped	2	3		4
	It would have been the same without them				
	They disturbed				
	They greatly disturbed				
Total number of answers		2	3	2	6
Total number of participants		2	3	3	6

Table 35: Case Study 2 - Questionnaire, Question 1

Welche Informationen aus den Diagrammen waren besonders hilfreich / wichtig für dich? Warum? **(Which information in the diagrams was especially useful / important for you? Why?)**	
4-April	Nützlich waren für mich vor allem die Prozessinformationen, die mit dem „reinen“ technischen Elise-System nicht abgebildet werden können und wichtige Hintergrundinformationen lieferten. Z.B. wie die Bib-Abläufe sind, vorläufige Verfahren, was passiert , wenn ich Literatur haben möchte, die Bib aber schon im Status „working“ ist. Diese Dinge kann man zwar auch erklären, sie sind jedoch in der Grafik präsenter. Als es um die konkrete Handhabung des technischen Systems Elise ging, traten die Diagramme in den Hintergrund. Da war war das System selbst

Welche Informationen aus den Diagrammen waren besonders hilfreich / wichtig für dich? Warum? **(Which information in the diagrams was especially useful / important for you? Why?)**	
	wieder anschaulicher.
	Besonders hilfreich waren die Diagramme, um den Ablauf der Nutzung im Vorhinein sehen zu können. Dadurch war klar, wie Elise zu benutzen sein würde. Gut finde ich auch, dass der gesamte Prozess zu sehen ist, also insbesondere auch die Arbeit der Bib. So kann ich sehen, in welcher Form ein Artikel besorgt werden kann und wie er dann zur Verfügung stehen wird. Bezüglich der Stati eines IVZ macht das Diagramm zur Nachbestellung klar, was diese bedeuten.
12-April	Man konnte gut die einzelnen Verbindungen untereinander und die Rangfolge (besser: Abarbeitungsreihenfolge) erkennen. Bei Neuerungen ist es immer ganz gut zu wissen, wo man steht, bzw. was von einem erwartet wird!
	Nützlich fand ich die Informationen, die deutlich gemacht haben, wie das Zusammenspiel zwischen Bib und Wimi-Aktivitäten funktioniert. So konnte ich darstellen, was die Bib macht und gleichzeitig transparent machen, was das wiederum für Auswirkungen auf die Wimis hat und umgekehrt..
	Da wir wechselseitig die Diagramme und die Oberfläche von Elise für die Schulung verwendet haben, haben die Diagramme besonders geholfen, bestimmte Interaktionen mit Elise zu motivieren. Der im Diagramm dargestellte Prozess der Nutzung von Elise hat an einigen Stellen bedingt, dass wir uns Elise angeschaut haben. Gleichzeitig konnte die Nutzung so auch in den Gesamtprozess eingegliedert werden. Am Ende der Schulung war Alex so auch klar, wie Wissenschaftler und Bib-Team Elise nutzen sollen/wollen und der simulierte Durchlauf einer Bestellung hat auch reibungslos funktioniert.
19-April	Auf Detailnachfragen konnte man sehr gezielt und schnell anhand der Diagramme überschauen, ob der angesprochene Aspekt bereits modelliert wurde, noch vereinbart wurde oder bisher gar nicht berücksichtigt wurde.
	Die einzelnen Aktivitäten und wer was macht! Die Frage der Weiterverarbeitung einmal besorgte Artikel konnte diskutiert werden
26-April	Kommentare zu Aspekten, die schon in anderen Sitzungen vorbesprochen wurden. Sie waren ein guter Erinnerungsanker.
	Ich kann mich weniger an die konkreten Inhalten der Diagramme erinnern, außer die Rollen, Bib, Gabriele und IMTM-Staff, sondern eher an die "Ansicht" der Diagramme. d.h. ich sehe einen Prozess in meinem kopf vor mir wie das techn. System Elise demnächst von uns "bedient" werden soll
	Sequenzierung der Schritte, beteiligte Systemtypen und deren Wechsel während der Bearbeitung
	Offensichtlich haben die Diagramme heute dazu geführt, dass alle Anwesenden so weit "im System waren", dass wir (wie auch in der Sitzung angemerkt) mehr über Weiterentwicklung geredet haben, als über die Einführung. Fragen wie "Kann man das nicht anders machen, wenn ich dies und das machen möchte" hatte ich eigentlich weniger erwartet.
	Immer dann, wenn ich als User was machen sollte, war meine Aufmerksamkeit besonders hoch. Die technische Umsetzung warum was nicht, nur, schwer oder gar nicht geht könnte ausgeblendet werden, wenn die Zielsetzung der Sitzung der Frage des Zeitpunktes der Einführung dient und nicht der Verbesserung des Systems.
	Ich hab natürlich geguckt, welche Änderungen meine Anmerkungen letztes mal bewirkt haben (leider war die Rollenzuordnung IMTM Team nicht geändert) Ich habe allgemein gemerkt, dass mir die Diagramm helfen, falls ich mal nicht konzentriert war, das bis zu einem gegebenen Zeitpunkt gesagt am Diagramm

Welche Informationen aus den Diagrammen waren besonders hilfreich / wichtig für dich? Warum? **(Which information in the diagrams was especially useful / important for you? Why?)**	
	nachzuvollziehen und notfalls noch mal nachzufragen – Highlighting hätte übrigens geholfen, Aufmerksamkeit zu binden (aber bitte ohne Attribut-Highlighting)

Table 36: Case Study 2 - Questionnaire, Question 2

Gab es Probleme hinsichtlich der Nützlichkeit der Diagramme, die durch eine andere Form der Dokumentation (bspw. Text, Freihand-Zeichnungen) hätten vermieden werden können? Wenn ja, welche? **(Were there problems regarding the usefulness of the diagrams which could have been avoided by other forms of documentation (e.g. text, free-hand-drawings)?)**	
4-April	Als ich mich auf das Treffen vorbereitet habe, fehlten mir an manchen Stellen weitergehende/ergänzende Erläuterungen zu den verwendeten Begriffen: Mir war nach der längeren Pause oft nicht mehr klar, was einzelne Begriffe konkret bedeuteten. Durch die Modellierungssprache, die den Diagrammen zu Grunde liegt, konnte ich mich auch nach der längeren Pause wieder gut in die vereinbarten Prozesse einarbeiten. Freihandzeichnungen ohne speziell festgelegte Notation hätte dies erschwert. Die Verwendung des Seeme-Editors gewährleistet darüber hinaus eine gewisse „Klarheit“ oder „Ordentlichkeit“ der Diagramme, die ich persönlich sehr schätze als Freihandzeichungen. Eine textuelle Beschreibung des Prozesses hätte ich zu aufwendig (auch in der Rezeption) gefunden. Um den beschriebenen Text zu kapieren, hätte ic wahrscheinlich versucht, diesen wieder als Diagramm zu modellieren. Das wäre m.E. keineswegs besser gewesen, sondern schlechter.
	Mein größtes Problem beim Testen vorher war, dass ich den Benutzernamen nicht angegeben hatte, dies wird aus dem Diagramm auch nicht klar. Auf die Darstellungs_form_ würde ich das aber nicht zurückführen. Hilfreich wären evtl. an einigen Stellen Screenshots der Masken, um die Bearbeitungsschritte genauer beschreiben zu können.
12-April	Das Springen zwischen den einzelnen Diagrammen war anfänglich etwas verwirrend, da nicht sofort ersichtlich war, warum jetzt gewechselt wurde. Einige Diagramme waren nur Ausschnitte aus dem großen. Im nachhinein war aber deutlich, dass man im großen Diagramm unmöglich noch weiter in Detail hätte gehen können, ohne das es unübersichtlich geworden wäre. Im Großen und Ganzen war die Dokumentation logisch und nachvollziehbar und mir fällt keine bessere Möglichkeit ein.
	Um Anregungen, Probleme zu notieren, bei denen mir nicht sofort klar war, wo ich sie im Diagramm verorten soll, war mir Freitext lieber. So hatte ich wenigstens eine Notiz und kann später genauer überlegen, an welcher Stelle im Diagramm ich sie kenntlich machen möchte. Vielleicht fehlt so den Textanmerkungen aber auch wieder der Kontext, wenn ich sie wieder im Seeme verorten möchte.
	Keine
19-April	Nein
	NEIN
26-April	m.E. nicht
	Wenn überhaupt, dann nur ergänzend, bspw. Screenshots von ELise, da es so schlecht zu lesen war am BEamer. Oder Hilfestellungen zur "Anmeldung", ca. eine Seite dazu als handout

Gab es Probleme hinsichtlich der Nützlichkeit der Diagramme, die durch eine andere Form der Dokumentation (bspw. Text, Freihand-Zeichnungen) hätten vermieden werden können? Wenn ja, welche? **(Were there problems regarding the usefulness of the diagrams which could have been avoided by other forms of documentation (e.g. text, free-hand-drawings)?)**	
	Syntaxdiskussion (" Das Diagramm entspricht nicht dem, was Du sagst..?") kann man als abgleiten empfinden. Ist aber andererseits auch von Vorteil, wenn man dadurch diskrepanzen zwischen Verständnissen aufdeckt.
	-
	Ich habe einige Verknüpfungen nicht verstanden: Pfeile in bestimmter Richtung, Rollen ohne Verbindungen. Bei einigen Entitäten war mir nicht klar was, was ist. Vollständige Handlungsvollzüge, paradigmatisch dargestellt, gefallen mir generell besser. Beispiel: Hier ist die Liste alt, jetzt ist die Liste so etc.. jetzt will ich einen Artikel haben dann muss ich das machen. Es wurde zu sehr die Entwicklerperspektive eingenommen, den DAU interessiert nicht ob und wer eine Liste von Hand einstellt.
	Da der Prototyp so schlecht zu sehen war, wusste man nie genau wo man ist – ein paralleles Highlighting auf dem SeeMe-Diagramm hätte geholfen – also Zuordnung zu Maske zueingebetteter Entität. Ich glaube doch, dass wir Diagramme hätten haben sollen, die bis auf den einzelnen Dialogschritt runtergebrochen die Schulung unterstützen sollten.!?

Table 37: Case Study 2 - Questionnaire, Question 3

Sonstige Anmerkungen?? **(Any other remarks??)**	
12-April	Was ich wirklich störend fand, war die noch unzureichende Präsentation der Diagramme. Zum einen war ich mit den Seemes noch nicht so vertraut, als das mir immer sofort klar war, wo welcher Prozessschritt, auf den ich springen wollte, steht. Da die Diagramme nicht komplett auf eine Bildschirmseite passen, d.h. immer etwas abgeschnitten sind, erschwert das die Orientierung. Der Moderator muss sich schon sehr gut mit den Diagrammen auskennen, um damit gut „jonglieren“ zu können. Da war ich nicht genug vorbereitet. Vielleicht wäre es auch gut, ein komplettes Seeme als Poster zu haben, einen speziellen SeeMe-Ausschnitt auf einem Beamer und das Diagramm auf dem anderen.
19-April	Der Fragebogen hätte direkt nach der Sitzung ausgeteilt werden können mit der Bitte um Beantwortung – ich kann mich jetzt kaum noch daran erinnern.
26-April	Während ich in den anderen Sitzungen die Kombination von seeme-Diagrammen und Elise sehr gut fand, hatte ich in dieser Sitzung das Gefühl, mehr mit Elise gearbeitet zu haben, bzw. die Kombination der beiden Darstellungsformen nicht so gut kombinieren zu können. Dies lag m.E. daran, das ein Teil der Aufmerksamkeit durch die Moderation absorbiert wurde. Ansonsten fand ich es in den anderen Sitzungen besser, mit den Prozessschritten zu beginnen, die im Fokus der Teilnehmenden stehen (also Artikel auswählen) und die anderen Prozesse erst im Anschluss darzustellen.
	Es war insgesamt gut und ich hab verstanden, wie wir ELise demnächst nutzen werden. :-) Ich fand es schwierig, dass John so kleinigkeiten wie die Modellierung diskutiert. Er sagte, dass Carina nicht das gesagt hat, was im Modell zu sehen sei. Das ist für den "Kunden" uninteressant. Der Kunde verknüpft in dem Moment das, was die Moderatorin dazu sagt. SChließlich lernen die Kunden die Notationsregel nicht auswendig.

Sonstige Anmerkungen?? **(Any other remarks??)**	
	Ich fand es schwierig, so lange die Aufmerksamkeit auf die Diagramme zu halten. Aber da kann daran liegen, dass ich derzeit ziemlich gestresst bin. Ich hätte einen schnelleren oder iterativ kleineren WEchsel zu ELise besser gefunden, weil ich so neugierig war, wie das techn. System funktioniert. (Vielleicht war mir die Zielsetzung nicht ganz klar: diente es zur Schulung, bedienung und Umgang mit ELise? oder eher zur Diskussion, was wir im prozess noch anders machen sollten? Vermutlich sollte es eine Verknüpfung von beiden sein? Wenn es eher das zweitere war, dann dachte ich, dass wir dies bereits schon mal gemacht hatten.)
	Das Ziel der Sitzung war mir nicht 100% klar, (konnte es mir denken...) das kann aber auch ein meiner kurzen Abwesenheit zu Beginn gelegen haben. Das Wechselspiel zwischen den Medien hätte deutlicher werden können. Teilweise wurde an zwei Baustellen parallel herumgewerkelt. (Carina erklärt was im Diagramm, Phil sucht zur selben Zeit seinen Weg durch Elise ...
	Rückblickend finde ich interessant, dass die Diagramme eigentlich ein Ergebnis aus Sitzungen des Großteils der heute Anwesenden sind und ein paar Monate später offenbar so viele Änderungswünsche und Ergänzungen aufkamen, dass man als Außenstehender den Eindruck gewinnen konnte, es würde sich um einen von Dir vorgeschlagenen Prozess handeln, der nun angepasst werden sollte.
	Die Diskussion hat gezeigt, dass die wichtigen Fragen angesprochen und verstanden wurden. Der Einführung steht nichts mehr im Wege.

Table 38: Case Study 2 - Questionnaire, Question 4[36]

		26-Apr
How do you judge the usefulness of the form of the diagrams (i.e. SeeMe-Diagrams) of today's meeting?	It helped very much	2
	It helped	4
	Any other way would have been as good	
	It disturbed	
	It greatly disturbed	
Total number of answers		6
Total number of participants		6

Table 39: Case Study 2 - Questionnaire, additional question

[36] Names changed

11.6 Appendix – Statistics of Use for Case Study 2

For a period of about 6 weeks the usage of the CSCW-application ELISE in case study 2 was monitored and entered into a tables which are summarized in Table 40. For each table of content, the following in formation is given:

- Name of Journal and Issue
- The day on which it was published in ELISE (column "publ.")
- The number of scientists who read the table of contents (columns "R")
- The number of scientists who ordered one or more articles from the table of contents (columns "O")

The information was gathered six times, the dates are given in the first row.

		16-Jun		22-Jun		4-Jul		11-Jul		20-Jul		28-Jul	
Name and Issue of Journal	**Publ.**	**R**	**O**	**R**	**O**	**R**	**O**	**R**	**O**	**R**	**O**	**R**	**O**
International Journal of Human Computer Studies (2005) Band 62, Heft 4	24-May	8	0	8	0	8	0	8	0	8	0	8	0
IEEE Transactions on Systems Man and Cybernetics - Part A - Systems and Humans (2005) Band 35, Heft 3	24-May	8	0	8	0	8	0	8	0	8	0	8	0
WSI Mitteilungen - Wirtschafts und Sozialwissenschaftliches Institut (2005) Band 58, Heft 4	24-May	8	1	8	1	8	1	8	1	8	1	8	1
New Media and Society (2005) Band 7, Heft 3	24-May	8	4	8	4	8	4	8	4	8	4	8	4
Zeitschrift fur Arbeits und Organisationspsychologie (2005) Band 49, Heft 2	24-May	8	4	8	4	8	4	8	4	8	4	8	4
Harvard Business Review (2005) Band k.A., Heft REVIS	1-Jun	8	0	8	0	8	0	8	0	8	0	8	0
Information Society (2005) Band 21, Heft 3	1-Jun	8	4	8	4	8	4	8	4	8	4	8	4
Requirements Engineering (2005) Band 10, Heft 2	13-Jun	1	1	3	2	8	3	8	3	8	3	8	3
Harvard Business Review (2005) Band k.A., Heft 6	13-Jun			3	2	8	2	8	2	8	2	8	2
International Journal of Cooperative Information Systems (2005) Band 14, Heft 2	13-Jun			3	1	8	2	8	2	8	2	8	2
International Journal of Human Computer Interaction (2005) Band 18, Heft 3	13-Jun			3	2	8	5	8	5	8	5	8	5
Spektrum der Wissenschaft (2005) Band k.A., Heft 6	13-Jun			3	0	8	0	8	0	8	0	8	0
WSI Mitteilungen - Wirtschafts und Sozialwissenschaftliches Institut (2005) Band 58, Heft 5	13-Jun			3	0	8	1	8	1	8	1	8	1
Information Knowledge Systems Management (2004) Band 4, Heft 3	16-Jun			3	0	8	0	8	0	8	0	8	0
International Journal of Human Computer Studies (2005) Band 62, Heft 5	16-Jun			3	0	8	1	8	1	8	1	8	1
International Journal of Human Computer Studies (2005) Band 62, Heft 6	16-Jun			3	1	8	1	8	1	8	1	8	1
International Journal of Human Computer Studies (2005) Band 63, Heft 1	16-Jun			3	1	8	2	8	2	8	2	8	2
ACM Transactions on Information Systems - TOIS (2005) Band 23, Heft 2	22-Jun					6	0	7	0	8	0	8	0
Empirical Software Engineering (2005) Band 10, Heft 3	22-Jun					6	2	7	3	8	4	8	4
Zeitschrift fur Soziologie (2005) Band 34, Heft 2	22-Jun					6	1	7	1	8	2	8	2
British Journal of Social Psychology (2005) Band 44, Heft 2	4-Jul							2	0	8	1	8	1

		16-Jun		22-Jun		4-Jul		11-Jul		20-Jul		28-Jul	
Name and Issue of Journal	**Publ.**	**R**	**O**	**R**	**O**	**R**	**O**	**R**	**O**	**R**	**O**	**R**	**O**
Communications of the ACM - Association for Computing Machinery - CACM (2005) Band 48, Heft 6	4-Jul							2	0	8	1	8	1
Computer und Recht (2005) Band 21, Heft 6	4-Jul							2	0	8	2	8	2
Datenschutz und Datensicherheit (2005) Band 29, Heft 6	4-Jul							2	0	8	1	8	1
IEEE Technology and Society (2005) Band 24, Heft 2	4-Jul							2	1	8	3	8	3
IEEE Transactions on Professional Communications (2005) Band 48, Heft 2	4-Jul							3	2	9	6	9	6
Informatik Spektrum (2005) Band 28, Heft 3	4-Jul							2	1	8	4	8	4
International Journal of Educational Research (2004) Band 41, Heft 2	4-Jul							1	0	8	1	8	1
IX - Magazin fur Professionelle Informationstechnik (2005) Band k.A., Heft 7	4-Jul							2	0	8	0	8	0
Journal of Science Education and Technology (2005) Band 14, Heft 2	4-Jul							2	0	8	0	8	0
Qualitative Inquiry (2005) Band 11, Heft 4	4-Jul							2	0	7	0	7	0
Zeitschrift Fuhrung und Organisation (2005) Band 74, Heft 3	4-Jul							2	0	7	4	7	4
Computer Supported Cooperative Work (2005) Band 14, Heft 3	11-Jul									7	2	7	2
Information and Software Technology (2005) Band 47, Heft 10	11-Jul									7	2	7	2
International Journal of Educational Research (2004) Band 41, Heft 1	11-Jul									7	1	7	1
Organization Science (2005) Band 16, Heft 3	11-Jul									7	2	7	2
Spektrum der Wissenschaft (2005) Band k.A., Heft 7	11-Jul									7	1	7	1
Harvard Business Review (2005) Band k.A., Heft k.A.	20-Jul											4	2
Information Knowledge Systems Management (2004) Band 4, Heft 4	20-Jul											4	0
New Media and Society (2005) Band 7, Heft 4	20-Jul											4	2
Qualitative Research (2005) Band 5, Heft 3	20-Jul											4	1
WSI Mitteilungen - Wirtschafts und Sozialwissenschaftliches Institut (2005) Band 58, Heft 6	20-Jul											4	0

Table 40: ELISE - Statistics of Use