electronically published at HICCS 2007 the Fortieth Annual Hawaii International Conference on Systems Sciences: http://www.hicss.hawaii.edu/diglib.htm
[this version includes minor modifications unlike the original publication]

Socio-technical Self-description as a Means for Projects of Introducing Computer Supported Cooperation

Thomas Herrmann; Gabriele Kunau; Kai-Uwe Loser
University of Bochum
Information and Technology Management; Institute for Applied Work Science
44780 Bochum, Germany
{thomas.herrmann, gabriele.kunau, kai-uwe.loser}@rub.de

Abstract

The concept of socio-technical self-descriptions can be employed to run CSCW-projects where several perspectives have to be integrated in a process of decision making. The decisions are related to technical as well as organizational structures and their interplay. Current approaches do not sufficiently match the challenge to facilitate the communication processes during CSCW-projects. The alignment of software-development and configuration on the one hand with organizational changes and regulations on the other hand need specific forms of documentation which can be considered as a kind of self-description. In this paper we explain how semi-structured diagrams can fulfill these requirements, if they are embedded in appropriately facilitated communication processes. We provide criteria and recommendations which were derived from several case studies and were refined during a project which planned to support the cooperation between truck drivers and dispatchers by using mobile pocketPCs.

1. Introduction

In this paper we concentrate on the methodical support for planning and documenting the interplay between technical functionality and change of work-processes in the field of computer supported cooperative work (CSCW). We consider this interplay as structural coupling (cf. Kunau [14, p. 46] referring to [16]) which is typical for the outcome of CSCW-projects: Both, the technical system as well as the organizational work processes adapt to each other and mirror the structure of each other; furthermore they influence but do not determine each other. These characteristics are caused by the specific organizational impact of CSCW systems, and they are not sufficiently taken into account by conventional approaches of software-engineering and IT-projects. The linear waterfall approaches do not apply to CSCW; instead, a continuous involvement of users and different stakeholders is necessary, who develop their expectations together with the technical system. Furthermore, the deployment phase – as far as it is included in a software-engineering project – usually underestimates the needs for organizational change and for the developing of appropriate conventions for CSCW-usage [18].This is also the problem with more iterative, evolutionary approaches to project management going back to Boehm's spiral model [3] and the usage of use-cases as suggested by [11]. The evolutionary models are much more adequate with respect to the gradual development of requirements in CSCW projects but they do not really comply with the phenomena of evolutionary usage [19] or appropriation [6] of CSCW-systems. It is also less appropriate to rely on the role of a "business designer who "*…determines the business workers and business entities to realize a business use case, and also how they work together to achieve the realization.* [13, p. 147]" The underlying assumption that work processes can be engineered is not compatible with our case experience as well as with theoretical insights about structural coupling.

To achieve a more feasible approach for CSCW-IT-projects we emphasize a concept of communicational intervention which respects the autonomy of the organizational system on the one hand and develops and documents the needed conventions on the other hand. This communicational intervention needs systematical facilitation as well as continuous documentation to support the engineering process of the CSCW-software. We propose to consider this highly interwoven, continuous process of communicating and documenting as a process of producing a socio-technical self-description (StSd) for the organizations which introduce CSCW. From our empirical work in several cases [10, 15] we suggest that the following aspects have to be dealt with in the self-description which is supported by the CSCW-project:

- the technically supported work processes have to be highly interwoven and continuously adapted to each other
- integration of multiple perspectives of the involved stakeholders and affected people ,
- the documentation has to support technical configuration as well as the coordination between workers. However, it has also to deal with incompleteness, since work can not be completely planned ahead and made visible [22]
- the documentation should be clear and specific enough so that the members of the organization are willing to consider it as a kind of self-description of their organization. Therefore, they should be able to recognize the documentation as a description which stems from inside their organization instead of coming from outside [7].

The research on socio-technical self-description [14] was based on our former work on a semi-structured modeling method (SeeMe [9]) and a facilitation method, the socio-technical walkthrough (STWT, [9]). The new insights presented in this paper derive categories of the self-descriptions' content as well as criteria how to detect this content and to represent it.

We have used semi-structured diagrams for the StSd which employed the modeling method SeeMe [9] SeeMe has been elaborated, evaluated and further refined through several action research projects [15] [10]

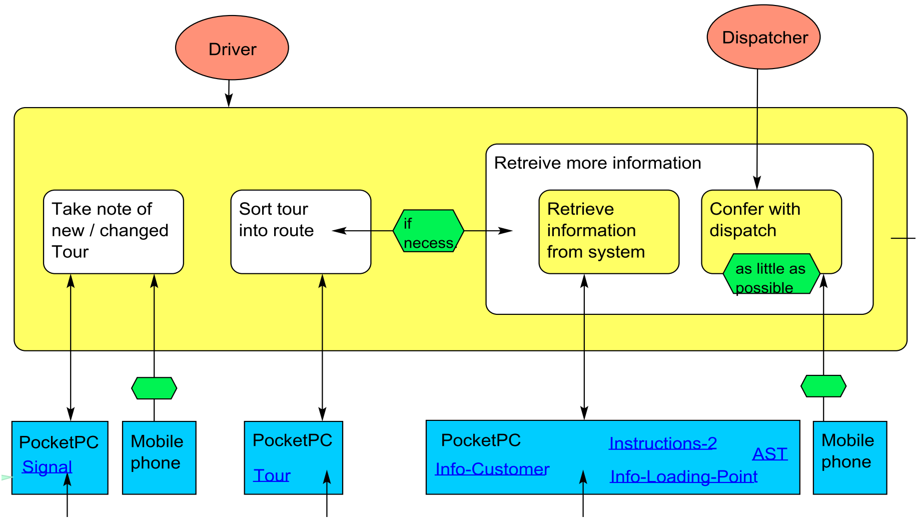

Fig. 1. "New Tour"

in CSCW related projects. We found that modeling notations such as Petri-net-diagrams, data-flow models, or entity relationship diagrams or UML were too specifically dedicated to the purpose that "*…members of the development team can unambiguously communicate their decisions to one another.*" [13, p. 12]." In our cases, not only the members of the development team had to be involved, and we had to accept that the documents were partially ambiguous. In [8] it is described how the STWT facilitates the communication processes and the organizational change. Our research resulted also in a tool – the SeeMe-Editor – for creating, presenting and jointly editing the diagrams. The facilitation method STWT ensures that the diagrammatic artifacts are embedded into a communicative context within an organization. STWT and the modeling method SeeMe were carefully compared [14] with other software-engineering and methods being relevant for CSCW-project design such as PD [12], JAD [5], scenario-based Design [4] Multiview [2]. Compared with these methods, SeeMe and STWT introduce new possibilities of handling incompleteness and multiple perspectives. Furthermore it became obvious that widespread methods in the area of software-engineering (such as UML, [20]) or business process modeling are not sufficient to express the specifics of socio-technical constellations. Especially the use-case approach is inappropriate if it pursues to *„define a firm boundary around the system"*[13, p. 99], instead of taking its interplay with work processes and social structures into account

The latest case-studies which have inspired our work on StSd dealt with:

1. The introduction of mobile communication devices for logistics services of steel delivery, which support the cooperation between truck drivers and dispatchers.
2. The development of technical support and the negotiation of conventions for a group of scientists who want to have mutual awareness of interesting literature which they select from scientific periodicals.
3. A medical practice conducting radiography for diagnosis and treatment where a patient information system was introduced to support the communication between

the administrative personnel and the doctors and in between these groups.

We conclude this introduction with the example of the 1st case study to make our background and motivation clear. A theoretical section follows that introduces the concept of self-description and its relations for IT-projects. Then the criteria for developing and using StSd in CSCW-projects are given and illustrated by examples from a case study. We conclude by outlining ideas for further research.

Case Study: Steel-Delivery

We briefly summarize the 1st case study in order to provide a proper background for the understanding of the examples. The project that constitutes the core of the case study had the goal of designing a mobile communication system supporting truck drivers and their dispatchers in a German logistics company. The system was intended to gradually replace the paper-based procedures not only between drivers and dispatchers but also between the logistics company, their customers and the recipients of the goods. For the drivers, a mobile device (PocketPC) was added to the equipment in their trucks; the dispatchers received an add-on to their office software. It was one of the goals of the project to integrate the technical design with the planning of new work-procedures that make expedient use of the mobile communication system. A participatory project was established which included the following stakeholders: drivers, dispatchers, local management, head office management, software-engineers. They represented different perspectives with respect to their interests, tasks, expertise, and power relations. The power relations between drivers and dispatchers are especially interesting: Although the dispatchers were holding the higher hierarchical position, the success of the tours was also strongly depending on the drivers decisions. In the course of the project, another aspect had to be explicitly taken into account for developing a self-description: It is the dispatcher whose workload will be significantly reduced if the new system is well established; but it is the driver who has to enter additional data into the mobile device in order to make the whole system work.

To run such a project, where multiple perspectives and interests had to be integrated, we conducted a series of workshops where semi-structured diagrams were used, to make different positions visible without getting lost in an information overflow as it can be caused if the completeness of specifications is formally required. One reason for conducting workshops in which drivers and dispatchers discussed the new system with each other and with the management was, to give everybody a chance to learn about the others' workplace, work-processes and specific problems; another reason was to make transparent how both contributed to the overall business processes. The diagrams were a means to capture the multiple facets of the discussion.

During the case study, 10 workshops following the facilitation method STWT [8] were conducted resulting in about 25 semi-structured diagrams. Fig. 1., Fig. 3., Fig. 4., and Fig. 6. are examples taken from this case study.

The case study can be considered as action research since we attempted to support the project in the role of preparing, facilitating and documenting the workshops, and we were systematically analyzing the content of the diagrams and the process of creating and modifying them from the theoretical perspective of StSd.

2. Theoretical Background of socio-technical self-description

The concept of self-description was inspired by the type of systems theory as it is elaborated by the German sociologist Niklas Luhmann to explain how organizations can create and maintain their identity [16]. Self-descriptions serve two purposes in this context: First, they guarantee stability by defining the boundaries of an organization. Second, self-descriptions enable change by being subject to reflection and debate. Examples for explicit organizational self-descriptions include: an organizational chart that describes who belongs to which department; an organization's mission statement which defines values that (should) guide the behavior within the organization; an ISO9001 process description which describes the expectations how certain tasks need to be carried out.

The deployment of a new system by a CSCW-project requires that an organization has to partially change its work processes and conventions, and subsequently to adapt its self-description. We call this "socio-technical" self-descriptions (StSd) because they combine social and technical aspects and they help to form a unit which is often referred to as socio-technical system. StSd serve the special purpose of explicitly describing how the use of a CSCW-system is integrated into the organization's work procedures [14].

Fig. 1. is an example of a StSd from the 1st case study: the diagram describes work-procedures of drivers and dispatchers when new tours are entered into the system; this includes the coordination between them as well as the usage of specific functionality of the mobile communication system, depicted as *PocketPC.*

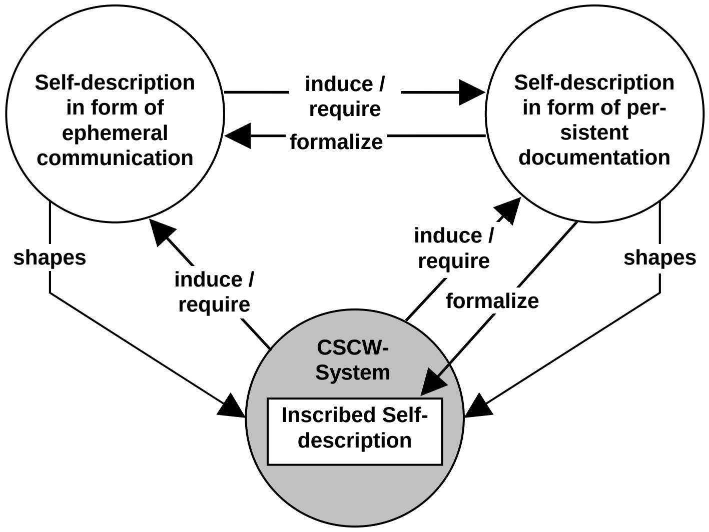
Self-description in form of ephemeral communication
induce / require
formalize
Self-description in form of persistent documentation
shapes
induce / require
induce / require
formalize
shapes
CSCW-System
Inscribed Self-description

Fig. 2. Three Forms of Socio-Technical Self-Descriptions

It is important to note, that socio-technical self-descriptions do not exist in form of homogeneous, canonical documents. Rather, they are distributed over three levels as illustrated in Fig. 2. In organizations one can distinguish between formal and informal self-descriptions; this distinction often coincides with the difference between oral and documented self-descriptions. An unwritten tradition – that the truck drivers call their dispatchers in the evening to tell them that the last tour is completed – is an example of such an informal self-description. In the diagram in Fig. 2., this phenomenon is referred to as *self-description in form of ephemeral communications*. The examples of organizational self-descriptions given above or the diagrams shown in this paper, belong into the category *self-description in form of persistent documentation.* This kind of documentation can be available as written text, graphical drawings, rough sketches etc. but also other kinds of artifacts such as parts of prototypes, mock-ups or even the setting of a room etc. serve this purpose. *Persistent documentation* should not be confused with formal specifications since it usually provides a mixture between formal and informal aspects.

Organizations using CSCW-systems have one additional type of self-description: those that are inscribed into the CSCW-system. Any CSCW-system contains aspects of the organization’s self-description; these may be contained in the hierarchy and access rights of folders containing documents; in the contents and structures of electronic forms; in the sequence of actions in workflow management systems; etc. However, self-description is not only contained in the system but also gives instructions how the system has to be shaped.

The arrows between the three forms of self-descriptions indicate that there is an interchange between them. A self-description that first exists only informally may be formalized and become part of an official document. Similarly, the statements in a document may be implemented as features of a CSCW-system. The other way around, a formal agreement always needs informal agreements into which it is embedded. The arrows labeled *Induce / Require* indicate this relation. In this article we concentrate on *Self-description in form of written or graphical documentation*. It can be considered as reasonable that a CSCW-project attempts to establish a self-description referring to all three dimensions and their relationships of fig. 2.

With regard to the technical CSCW-system, the outcome of a project is documented by the features of the system itself. But how can we keep record of the discussions and contributions regarding the organizational aspects and the usage of the CSCW-system? Their traces are much more ephemeral and are therefore harder to track. In our case studies [9], semi-structured diagrams have proven to be able to serve this purpose. In order to make organizational decisions concerning the usage of a CSCW-system visible for the varying roles involved in the project, the decisions need to be documented in a way that all participants can recognize their contributions.

As it becomes clear with the work of Suchman [21][22], it should not be neglected, that StSd can never completely or exactly anticipate the future of the real usage of a software system. However, Suchman also claims that the more the descriptions stem from insiders the less stereotyped they are. CSCW projects have also to be supported by means such as prototypes, experience of other companies etc. The StSd, which always should explicitly refer to these means, can be considered as an important concept being worth to be methodologically investigated and supported. In the following section criteria are discussed of how StSd can be developed and used in the course of projects.

3. Criteria for StSd in CSCW-projects

One important task in the course of CSCW-projects is to facilitate meetings where the decisions for the software-requirements and for the organizational regulations have to be made. The socio-technical walk-through method suggests that the facilitator focuses the discussions during workshops by using questions which are related to certain topics and content. The work of the facilitator – as an important role of the IT-project – should be guided by certain criteria.

3.1 Categories of Content

Based on the empirical case studies we have identified six categories of content of high relevance. They were derived by an ex-post analysis of the diagrams stemming from the first two case studies mentioned in the introduction. Within both cases it was confirmed by questionnaires that the participants considered the contents of the diagrams as helpful [14]. The following categories were identified:

a) Work Procedures
b) Relations between work-related activities and software functions
c) Agreements concerning coordination of the primary task
d) Agreements concerning the usage of the CSCW-system
e) Usage of additional technical systems
f) Meta-level comments concerning the project.

We recommend that in any CSCW-project all six categories should be closely considered.

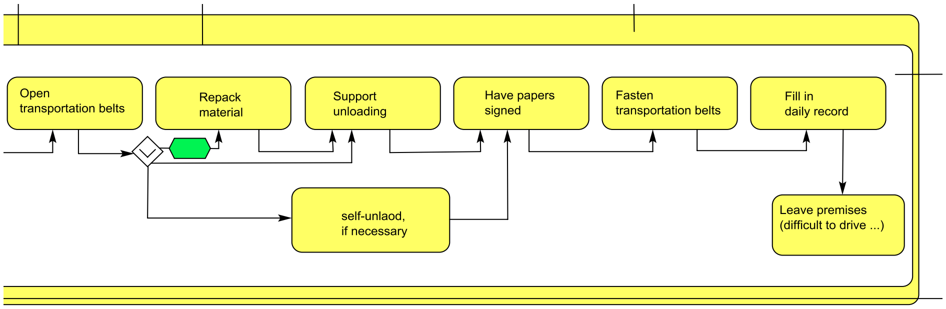
Open transportation belts
Repack material
Support unloading
Have papers signed
Fasten transportation belts
Fill in daily record
self-unlaod, if necessary
Leave premises (difficult to drive ...)

Fig. 3. Work-Procedures

a) Work Procedures

The work procedures provide the context into which the CSCW-system is to be embedded. Fig.3 is a detail from a large diagram and illustrates the drivers' activities when unloading their trucks at the customers' site.

Especially in early phases of projects, all tasks that seem to be of importance to the participants should be included into the diagrams, since they serve as a self-description. Our experience from the case studies shows that the description of the work-procedures will change in the course of the project. Activities will loose or gain importance as the CSCW-system develops; and accordingly activities will be eliminated (e.g. *Open transportation belts* and *Repack material, Fig. 3)* or added, or modified. The activity *Fasten transportation belts* was changed into the more general task of *Put truck into roadworthy state.* Representing this task, as Fig. 4. shows, was important, although it is not linked to an interaction with the system. The participants had discussed whether or not the departure from the recipient's site could be automatically signaled whenever the delivery note has been marked as signed. But then they decided that the task of *Put truck into roadworthy state* can be so time-consuming that it is worth noting it. But for this it is important that all drivers know that signaling the departure is really the last action at a recipient's site. This is an example that it is important to care systematically about the experience of every related role, also if some aspects are not directly linked with the system. Intensive reflection of the work procedure is an important success factor to ensure that the work after the CSCW-project is carried out with the system instead of around the system.

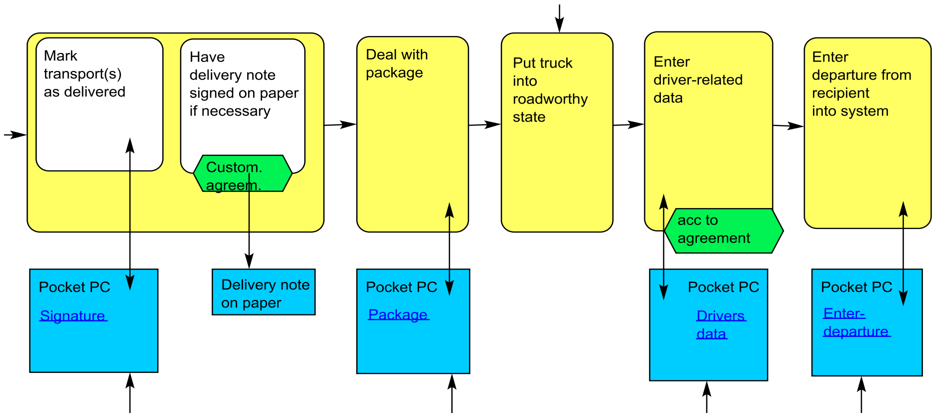
Mark transport(s) as delivered
Have delivery note signed on paper if necessary
Custom. agreem.
Deal with package
Put truck into roadworthy state
Enter driver-related data
acc to agreement
Enter departure from recipient into system
Pocket PC
Signature
Delivery note on paper
Pocket PC
Package
Pocket PC
Drivers data
Pocket PC
Enter-departure

Fig. 4. Documenting the Usage of *PocketPC*

b) Relations between work-related activities and software-functions

The second category of content describes how work-related activities relate to the CSCW-system. Fig. 1. and Fig. 4. provide examples from the case studies.

When relating work-procedures to functions of the CSCW-system, one should take care not to include too many new tasks that serve the sole purpose of handling the CSCW-system. With the modeling method SeeMe, it can be indicated that activities are only incompletely represented (e.g. by a semi-circle added to diagram elements). This can be used to mark those parts of the diagrams which should be specified in detail afterwards. StSd are no sequence diagrams as they are known in the interaction view of the UML [19]. In the case studies we included new tasks for handling the CSCW-system in those cases when the CSCW-system did indeed give rise to an additional step in the work-procedures. Examples are the tasks *Register arrival* in Fig. 7. or *enter driver-related data* in Fig. 4. In other cases such as *Take note of new / changed tour* in Fig. 1. or *Deal with package* in Fig. 4. the driver's task with regard to his work-procedure is described and the *PocketPC* is linked as an (additional) resource.

It is important to document the conditions under which a certain software-function will be used. This recommendation relates to the agreements which are necessary for a coordinated use of a CSCW-system (see below). Fig. 5. provides an example: The dispatchers and drivers agreed that drivers should signal their arrival at a customer's site immediately upon arrival. This implies that the dispatcher is informed and that the corresponding workflow is started even if the driver might be unable to complete the delivery e.g. because no one is available to help unloading or because the customer refuses the acceptance of the goods.

c) Agreements concerning Coordination of Primary Task

The design and deployment of a CSCW-system influence the way of how work is organized and coordinated; and this does not only apply to tasks which are directly related to the usage of the CSCW system. Fig. 1. includes an example describing the cooperation between driver and dispatcher in the event of new tours. The driver is responsible to *Take note of new / changed tour*, he then *Sorts tour into route*. If necessary, he will *retrieve more information* and one way to do this is to *Confer with dispatch*. This sequence contains two agreements that were extensively discussed and then documented during workshops: First, the fact that the driver is responsible for sorting a tour (i.e. an unordered list of recipients) into a route (i.e. a list giving the chronological order in which he will deliver the goods). Second, the agreement, that additional consultation with the dispatchers should be reduced to a minimum. The latter agreement was important because one of the problems to be solved by the deployment of the mobile communication system was that frequent phone calls from the drivers interrupted the dispatchers in their complex task of planning the tours for the next day. It became obvious that facilitation of communication and the usage of diagrams are an important means to document the need for negotiation.

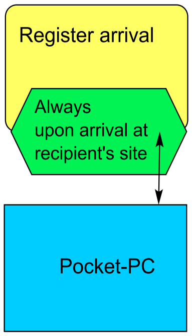

Fig. 5.

d) Agreements concerning the Usage of the CSCW-System

Our case studies confirmed findings, that the coordinated usage of a CSCW-system requires further conventions to be agreed upon by the users [18]. Therefore StSd include the documentation of agreements concerning the usage of the CSCW-system. Fig. 5 provides a typical example. The discussion in a workshop revealed that there were several options of using the function *Register arrival*. Some participants suggested using the function only once the driver can foresee that the process of delivery will be successful, otherwise the connected workflow should not be started. Others argued that it is important to register each arrival at a customer's site, even if delivery would not be successful. In the latter case the documentation would be especially important because the costs of a second attempt could be charged to the recipient. In some cases it might not be possible to find a decision on which all

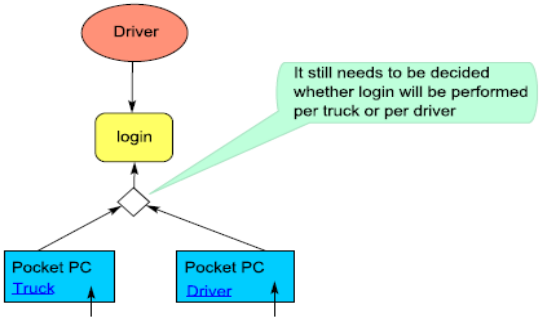
Driver
login
It still needs to be decided whether login will be performed per truck or per driver
Pocket PC
Truck
Pocket PC
Driver

Fig. 6: Metalevel comments

parties will agree. As it is shown in Fig. 6 it can be reasonable to document this situation in the diagram. From this documentation, the software-engineer derived the design rationale that it is reasonable to provide two options while the organizers had a reminder for a pending decision.

Supporting the StSd in parallel with the design and implementation of the technical CSCW-system allows these discussions to take place before the system is actually used. One advantage is that the results of these discussions may still influence the technical design if necessary; another is that the phase of actual deployment is smoother because the participants already share basic ideas about the usage of the new system.

e) Usage of additional technical Systems

In most organizations, the new CSCW-system will be deployed in the context of other resources such as IT-systems that may functionality which will overlap with the features of the new system. Since StSd should support the structural coupling of the organization with a new CSCW-system, it is important to explicate how the new CSCW-system will fit into the existing resources. In our case studies, the two most important existing resources were paper documents and mobile phones. Before the mobile communication system is deployed, paper documents are the main means for coordination between dispatchers, drivers and recipients. Fig. 4. documents that customers still may sign a *Delivery note on paper* if that has been agreed. With respect to the joint consideration of new as well as old technology and their interplay, the StSd-concept supports a leader of a CSCW-project more sufficiently than other methods which we found in the context of CSCW-design (cf. section 1).

f) Meta-level Comments Concerning the Project

The last category of content that was identified in the case studies contains meta-level comments concerning the project. Fig. 6. provides an example. During the case study, the diagrams were used for validating the technical prototypes as well as the intended work procedures. The participating software-engineers took note of all decisions directly related to their work; which mainly included accepted change requests for the prototypes. What was missing was a place to document ideas, problems or open issues that were not always directly linked to the implementation of the mobile communication system. Thus the diagrams are a way to document those aspects of the discussion of computer supported work-procedures that are regarded as important by the participants but that are not necessarily expressing the final technical specifications.

3.2 How to Know what to Model and when?

A facilitator of workshops within a CSCW-project can support a group in filling these abstract categories of content with information relevant for a specific project. One method is posing the right questions. Facilitating STWT-Workshops which support an organizational unit's development of a StSd requires preparing a set of questions from which the facilitator can flexibly select [8]. Using these questions for group meetings should not be confused with the document analysis or the interviews which usually should take place before the workshop to explore the field of the CSCW usage. Here a sub-set of questions which can be used to drive the creation and modification of diagrammatic models is given. Starting questions are typically: "With which action does the work process usually start?" or "How do you react to typical events which start the work procedure?" Most of our modeling workshops were driven by activity oriented questions – however, questions can also focus on the "most important document, which connects the different activities of the work process" or on the role "which is in charge of the majority of the relevant activities".

If a starting point is found – e.g. an activity Y – it is reasonable to go on with a kind of multi-dimensional chain-building such as:

1. Which activities have to take place so that this activity Y can be initiated, and do these prerequisite-activities have to be part of the model?
2. Which activities will follow or be prepared by Y and is Y really necessary?
3. Who or which role will initiate and /or carry out Y?
4. Which objects or documents are needed to carry it out? What is produced by Y, and are the produced results needed by other activities?
5. On which conditions does the decision to carry out an activity depend? Who will make the decision – and which conventions do need to be agreed upon so that the sequence of activities and the employing of resources lead to mutually accepted result?

These kind of "flow-oriented" questions are derived from interdependencies as they are outlined in coordination theory approaches such as described by Malone and Crowston [17]. We found that these questions are a necessary basis, but have to be completed by the socio-technical perspective of StSd. Therefore we propose additional questions which aim at the relationship between technology and work activities:

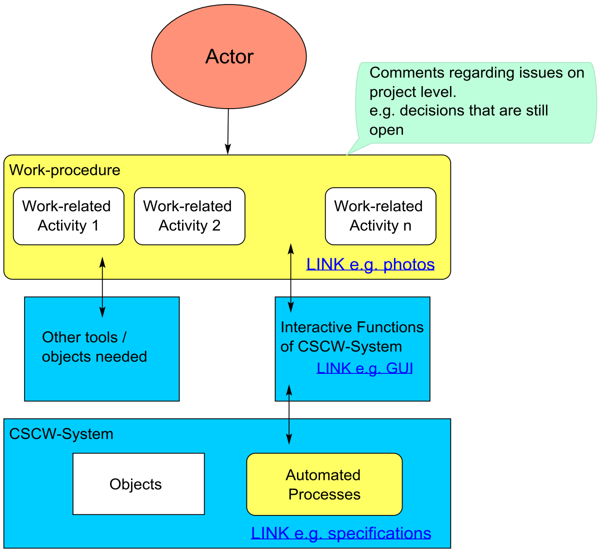

Fig. 7. Template for Socio-Technical Self-Descriptions

1. How is an activity – or should it be – technically supported? Is the activity a subject of human-computer-interaction or is it completely automatically controlled? Is it initiated by software or by human decisions?
2. Who configures the software which controls or initiates the activity? By which activities or roles is this configuration influenced?
3. Which documents are routed by the CSCW-system, which information is provided by the system and what kind of objects should or can be modified by computer based information processing?

All the answers which are obtained from these questions trigger a meta-question: How far or detailed should the answer be documented within the StSd and how should it be indicated that some information is left out (cf. 3.4) If parts of the diagram remain incomplete, a final meta-question of the systematic development of StSd is: "How do we proceed with this lack of specification? Will it be completed of will it be left for ad-hoc decisions?""

3.3 A Template for Socio-Technical Self-Description

Now that the contents of StSd and the procedure of developing it have been discussed, the next question is how these contents are represented in semi-structured diagrams. A modeling method like SeeMe [9] grants freedom in organizing the diagrams. While this is advantageous for adapting the models to different needs, it also may pose a problem for people employing a modeling method for the first time. Therefore we condensed our experience from the case studies into a template which provides room for each of the categories of content discussed above. Fig. 7 illustrates this template.

The template comprises four layers: It recommends depicting the roles of the actors at the top of the diagrams and directly below of them, the work procedures which they conduct. The work-procedures can additionally 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 placed below the work-procedures to express a close relationship. They can or should be linked to artefacts like GUI screenshots of the prototypes. The internal functions and components describe what the user can expect from the CSCW-system. Therefore static elements that are stored by the system as well as automated processes should be documented at the bottom of the diagrams. 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.

3.4 Representation of Situated Action and Contingency: Making Incompleteness Visible

When using a (semi-)structured modeling method, it is important to leave traces in the StSd which refer to the contingency of human action at the workplace. Suchman coined the term "situated action" for the unpredictable ad-hoc decisions which are necessary to successfully coordinate office work [21]. It can be tempting to oversimplify the representation of work-procedures by suggesting regularities which do not exist. If such simplified diagrams are e.g. taken as input into the design of CSCW-systems, the system may in the end not meet the expectations of the users. Consider the example in Fig.8. Here a *clerk* is responsible to *double-check the accuracy* of a *Customer Form*. In the first diagrams a) and b) he uses the *Customer Form* and *Additional Information* to complete the task. In the second version b), he performs the task only under the condition *Customer unknown*. In the third version c), there are two hexagons specifying the conditions are left empty. Here vagueness is used to express situated actions. The *clerk* has options in two places: first, it is left to his discretion whether or not he performs the task of *double-check accuracy* at all. Second, he can choose whether or not he includes *Additional Information* in fulfilling the task. The fact that the hexagons

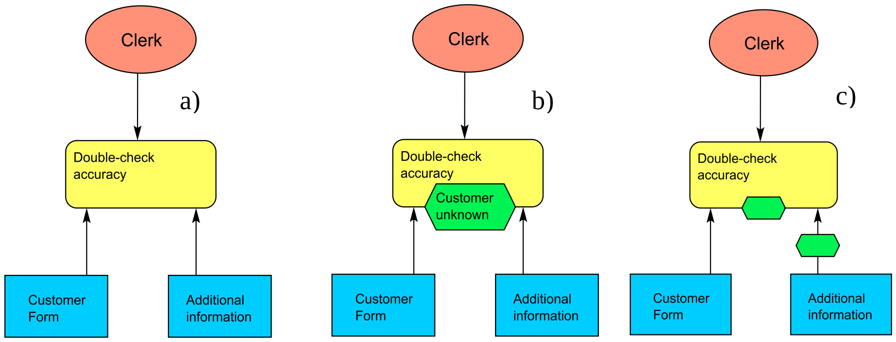

Fig. 8. Using Vagueness to Represent Situated Action

are empty, documents that there are no formalized rules that can be expressed at this point; it is indeed left to the *Clerk* and his judgment of a certain situation how the process is completed. Fig. 1. includes similar examples from the case study: whether or not the driver uses a *mobile phone* to contact the dispatcher is not determined by a fixed set of rules but rather by the driver's judgment of the situation.

But situatedness of action also applies to the usage of the CSCW-system itself. It cannot always be specified whether and under which conditions a CSCW-system will be used in a certain situation. Fig. 1. includes the example that the task *Retrieve information from system* is performed and that the *PocketPC* is used. But it is not specified which of the implemented resources the user chooses (here: *Info-Customer*, *Instructions-2*, *AST*, or *Info-Loading-Point*). In other cases like the task *Take note of new / changed tour* it is clear that a certain function from the *PocketPC* is to be used (here: *Signal*).

We suggest that an advanced modeling method should provide formal and informal means to indicate incomplete specification e.g. that a question is left unanswered. SeeMe demonstrates concepts of how to represent situated action as described in the examples above. These concepts to represent vagueness can also be applied within other modeling methods.

3.5 Integrating other Media

There are studies showing that abstract diagrams are a suitable means for supporting the communication with and among future users [15]. Nevertheless, we found it helpful to enrich the diagrams by interweaving additional material with hyperlinks. First, such additional material should help the participants associate the abstract diagram with their actual work-practice; second, as the CSCW-system is being implemented, it should build a bridge between the representation of work-procedures including additional agreements and the new software. As shown in Fig. 7, such links can occur on all levels of the diagrams. *Work-procedures* can be illustrated by photos from the work-place; and corresponding resources such as forms can be documented by photos or by PDF-documents. For the evaluation of prototypes it is helpful to integrate screenshots of the prototype with the diagrams. This way an integrated discussion of work-procedures and technical artifacts is supported (cp. [8] for a discussion on facilitation methods).

Since the diagrams serving as StSd are meant to support the communication among the roles being part of the prject rather than supporting the detailed technical specification of a software system, it is useful to shift detailed technical representations into separate documents which are then linked to the corresponding entities in the diagrams. The diagrams themselves should include descriptions of the CSCW-system that are understandable by the future users.

So far we have worked with additional visual material in form of pictures or documents; however any other type of media which is reachable by hyperlinks could also be included. All in all, the approach of linking material to the diagrams helps to make the integration of different perspectives more comprehensible.

4. Conclusion and Further Work

We have emphasized that CSCW-projects have to support the integration of the software system into the organizational structure of the work processes. This task cannot be solved by engineering methods, but needs a concept of intervention which combines facilitated communication with continuous documentation. We have explored how far the theoretically derived concept of socio-technical self description supports this approach. Therefore we have conducted two case studies on the basis of an action research approach. By a detailed analysis of the content and the evolution about 50 diagrams we have derived criteria which should be taken into consideration by StSD. StSDs in the context of CSCW-projects can be considered under the aspect of the process of their creation and modification, and they have a certain structure comprising ephemeral communication, persistent documentation and technical features mirroring structural aspect of the interplay between the software and the work processes. If STSD is used in future CSCW-projects we recommend to take the following aspects into account:

- it should provide feedback to the participants by mirroring the technical as well as organizational decisions and the coupling between them,
- it should be available after the decision processes and therefore be non-ephemeral by either being textually or graphically documented or by being an inscribed part of the technical system
- it should serve multiple perspectives as they are part of an organization's communication processes.
- it should be flexible enough to provide an appropriate mixture between complete, formal specification and incompleteness. There StSd should use a modeling method which includes possibilities to indicate vagueness as exemplified with SeeMe.
- it should be continuously available to be adapted if decisions have to be revised in the course of an organization's evolution [19]. While the design of CSCW-software might be widely completed with the end of a project, the design of the self-description is a continuous task.

So far we concentrated on the usage of StSd in the context of IT-projects. However, the concept of continuing design in the context of evolving use [1] [19] is well known in the field of CSCW. We assume that StSd in form of semi-structured diagrams are suitable artifacts to support an ongoing process of appropriation. But this aspect has not yet been in the focus of the case studies. Research questions related to this line of thoughts would be: How sustainable are StSds? Are they used beyond a specific IT-project? These questions should be picked up in future case studies.

REFERENCES

1. Andriessen' J.,H,.E.;, Hettinga' M. Wulf, V. (2003): Introduction to Special Issue on Evolving Use of Groupware. In: JCSCW, Vol. 12, No. 4, pp. 367-380.
2. Avison, David E.; Wood-Harper, A.T. (1990): Multiview: An Exploration in Information Systems Development. Oxford: Blackwell Scientific Publications.
3. Boehm, Barry W. (1988): A Spiral Model of Software Development and Enhancement. In: IEEE Computer May 1988. pp. 61 - 72.
4. Carroll, J. M. (Ed.) (1995): Scenariobased Design for human computer interaction. New York: John Wiley.
5. Crawford, A. (1994): Advancing Business Concepts in a JAD Workshop Setting. Yourdon Press, Englewood Cliffs, N.J.
6. Dourish, P. (2003): The appropriation of interactive technologies: some lessons from Placeless Documents. *Computer Supported Cooperative Work* 12, 465-490.
7. Grinter, R. E. (2000): Workflow Systems: Occasions for Success and Failure. In: *Computer Supported Cooperative Work* 9 (2000), 189-214.
8. Herrmann, T.; Kunau, G.; Loser, K.-U.; Menold, N. (2004): Sociotechnical Walkthrough: Designing Technology along Work Processes. In: *Proc. PDC 2004*, ACM, New York, 132-141.
9. Herrmann, T.; Hoffmann, M.; Kunau, G.; Loser, K.-U. (2004): A Modeling Method for the Development of Groupware Applications as Socio-Technical Systems. *Behaviour & Information Technology* 23, 2, 119-135.
10. Herrmann, T.; Hoffmann, M.; Kunau, G.; Loser, K.-U. (2002): Modelling Cooperative Work: Chances and Risks of Structuring. In: *Proc. COOP 2002*. Amsterdam: IOS Press, 53-70.
11. Jacobson, Ivar (1993): Object oriented software engineering. A use case driven approach. Wokingham: Addison-Wesley.
12. Kensing, F.; Simonsen, J.; Bødker, K. (1996): MUST – a Method for Participatory Design. In: J. Blomberg, F. Kensing, E.A. Dykstra-Erickson (Eds.): Proc. of the PDC 1996. Palo Alto, CA., 129-140.
13. Kruchten, Ph. (2004): The Rational Unified Process: An Introduction (Third Edition). Boston: Addison-Wesley.
14. Kunau, G. (2006): Facilitating Computer Supported Cooperative Work by Socio-Technical Self-Descriptions. Dissertation, University of Dortmund, (http://hdl.handle.net/2003/22226)
15. Loser, K.-U.; Herrmann, T. (2002): Enabling factors for participatory design of socio-technical systems with diagrams. In: *Proc. PDC 2002* Palo Alto: CPSR, 114-143.
16. Luhmann, N. (1995): Social Systems. Stanford, California: Stanford University Press.
17. Malone, T. W.; Crowston, K (1990): What is Coordination Theory and How Can it help design cooperative work Systems? In: *Proc. CSCW 1990*, 357-370.
18. Mark, G. (2002): Conventions and Commitments in Distributed CSCW Groups. In: *Computer Supported Cooperative Work* 11 (2002), 349 – 387.
19. Orlikowski, W. J. (1996): Improvising Organizational Transformation Over Time: A Situated Change Perspective. *Information Systems Research* 7, 1, 63-92.
20. Rumbaugh, J.; Jacobson, I.; Booch, G. (2005): The Unified Modelling Language Reference Manual, Second Edition. Addison-Wesley.
21. Suchman, L. (1987): Plans and situated actions: The problem of human-machine communication. Cambridge, U.K.: Cambridge University Press.
22. Suchman, L. (1995): Making Work Visible. *Communications of the ACM* 38, 9. 56-64.