Location as an Analysis and Design Metaphor in Work Settings with Complex Cooperations

Stefan Roock[a], Ingrid Wetzel[b], Henning Wolf[c], Heinz Züllighoven[d]
1roock@informatik.uni-hamburg.de[a]
wetzel@informatik.uni-hamburg.de[b]
1wolf@informatik.uni-hamburg.de[c]
zuelligh@informatik.uni-hamburg.de[d]
Software Engineering Group
Dept. of Computer Science
University of Hamburg
Vogt-Kölln-Straße 30
+49 40 5494 2306

Organizations are subdivided into units. The services or products the organization provides as a whole are performed by employees of different units in close cooperation. This cooperative work is supported by a sophisticated environment. Current software systems normally lack this aspect by providing system support merely on the perspective of separated single users applying the desktop metaphor.

This article claims that in work settings with complex cooperations the work environments need to be analyzed and captured in system design. The paper presents an approach using a wide range of different design documents for analyzing joint tasks in organizations and responsibilities and work environments of units. It uses the location metaphor for designing software systems capturing certain aspects of the real work environments.

Keywords: Cooperative Work, Organizational Development, Location, Analysis, Design, Metaphor, Room Metaphor, Design Document

Introduction and background

Traditionally, application software development has focused on the individual workplace. This was reflected by both the software product and its development process. This is also true for the tools and materials metaphor, in which we are rooted.

The tools and material metaphor provides a general guideline - a so called leitmotif: the workplace for expert-human work. With respect to this leitmotif developers are supported by two main analysis and design metaphors: tool and material. People use various kinds of tools to manipulate different materials in their everyday work. With this in mind, we try to identify tools and materials in the development process. After finding such tools and materials we build the system upon the notion of tool and material. While re-designing the existing physical materials in the future system, we do not transfer "copies" or simulations of the available tools. We design the software tools by taking advantage of the power and possibilities provided by computer systems. Therefore the software tools will share only essential concepts but few features with their physical counterparts. The metaphors are supplemented by a set of design documents and an evolutionary development process. The design documents are described in a following section while we only sketch the development process.

We first argue that the individual workplace approach and the well-known desktop metaphor are not adequate for analyzing and designing systems for supporting cooperative work tasks (section 2). In section 3 we describe the familiar document types used in the tools and materials approach together with new document types used to design cooperative systems. Based on these document types we will show in section 4 the central role of the notion of location and how cooperative systems can be built based on this notion. We will conclude, that the room metaphor is an adequate realization for locations in computer systems.

Problems with Software Development for Cooperative Work in Organizations

Cooperative work is work in organizations. The participants do their work in and for organizations. Organizational structures influence the ways people cooperate and vice versa. Software for cooperative work is software that will be embedded in cooperative work tasks in organizations. Such software will definitely change the ways people work and will influence organizational structures (for embedded software see Lehman, 1980). Therefore, developers can not analyze cooperative work without regarding the organizational structures. On the other hand software should not be developed with only these structures in mind, neglecting the individuals and their needs. With this interplay in mind, we can and should develop software and organizations with respect to each other.

The relation between the organization and the individuals on the one hand, and the relations between groups of individuals in organizations are very complex. They are not fully determined by the organizational structure. Our job as software developers is not to resolve the tension between the several actors. As a matter of fact we don’t believe that anyone in- or outside the organization can ever resolve this tension nor that this should be done. Only organizations with these tensions are living organizations and have, therefore, the possibility to adapt to market changes. Our job as software developers should be to moderate organizational development in the context of software development, which should involve both the concerned employees and the management as representative of the organization. This wide spectrum of participants with different backgrounds and qualifications requires simple and easy to understand document types. They should by no means overspecify the organizational structures nor are they meant to specify the requirements for the future software in a formal manner.

The tools and material metaphor ohas riginally focused on individual work and the workplace. It supports requirement analysis and software development with its metaphors and document types, e.g. scenarios, glossary and system visions (see Bäumer, Knoll, Gryczan, Züllighoven 1996). Recent work in the context of the tools and materials metaphor has pointed out the need for additional document types and design metaphors in order to analyze cooperative work and building suitable software. Wulf, Gryczan and Züllighoven 1996 introduced the term of process pattern for supporting asynchronous cooperative work tasks. We think of process patterns as a design metaphor rather than design documents and therefore present them in section 4. The notation of cooperation pictures and purpose tables for analyzing cooperations in organizations was provided by Krabbel, Ratuski and Wetzel (1996).

Document types for cooperative work

We use a document driven approach to software development. Scenarios, system visions and glossaries have been used successfully in analyzing and designing software for the individual workplace. They provide answers to the following crucial questions which arise during the development process:

- Which tasks are performed by the domain expert? (scenario)
- Which materials are used by the domain expert? (scenario)
- How does the domain expert work on these materials? (scenario)
- How does the domain expert perform the activities to fulfill a task? (scenario)
- How could the future software system look and how could it help the domain expert to perform his tasks? (system vision)
- What are the central terms and concepts of the domain, besides materials? (glossary)

These document types fall short when dealing with cooperative work tasks. They have to be supplemented by additional document types for cooperative work tasks, and not replaced, since the software should still support domain experts at their workplace. The additional questions that arise in the context of cooperative work are:

- What are the joint tasks in the organization? (ORC card[1], cooperation picture)
- Who are the participants of these joint tasks[2]? (ORC card, cooperation picture)
- What cooperations are performed to fulfill these joint tasks? (cooperation picture)
- Why are single coordination activities performed by the participants? (purpose table)
- Which are the locations for joint tasks? Where does the cooperation take place and what are the locations of the participants? (location picture)

ORC cards and cooperation pictures seem to overlap which they actually do. But we will see that ORC cards focus on all tasks of one participant, while cooperation pictures focus on the cooperations among all participants to perform a *single* or *few* joint tasks. Purpose tables provide important information about the effects of changes on the cooperational structures. Location pictures will aid us in designing the future cooperative software system. We will show later the special role of the location pictures for the design of the software system. All document types for cooperative work tasks can be used on different levels of abstraction. The participants of joint tasks can be viewed on the level of persons, roles, organizational units or even whole organizations. These levels can be mixed when appropriate. We use the term of *organization actor* to refer to such a participant.

We present a short overview of the central document types for analyzing and designing individual and cooperative work. Some of them are well established and crucial for the tools and material metaphor. These are scenarios, glossaries and system visions (see Bäumer, Knoll, Gryczan, Züllighoven 1996). Cooperation pictures and purpose tables have been introduced in the latest work of Krabbel, Ratuski, Wetzel (1996). We have supplemented the set with ORC cards and location pictures. The cooperation pictures were extended slightly.

We will describe the document types in the following pattern. We use this pattern since we believe that the main differences between the document types can be found in the pattern categories. The established document types are described briefly. We use the organization SOMCA (name changed) as an example. SOMCA produces chemicals in a one phase production process in which the necessary raw materials are mixed together.

---

[1] ORC is the acronym for **O**rganizational actor, **R**esponsibility, **C**ollaboration. ORC cards are somewhat similar to CRC cards. We will introduce ORC cards in chapter 3.

[2] The notion auf *joint task* was introduced by Krabbel, Ratuski, Wetzel (1996). A joint taks is performed by two or more persons together.

Description pattern for document types
Description: What type of documentis it?
Elements: What are the logical elements of the document and how are they presented?
Participants: How and by whom is the document created and used?
Purpose: What do we use the document type for? Which information does the document type contain? What is its role in the development process?
Relationships: How is the document type related to the other document types?

Scenario

Description: Scenarios describe the current work situation of individuals in words. Table 1 (identified materials are written *italic*) shows a sample scenario. Scenarios play a central role at the beginning of the development process.
Elements: A scenario consists of

- a name,
- the description and
- a context, in which the scenario takes place.

Participants: Developers write scenarios based on interviews with potential users. The written scenarios are reviewed by the interviewed persons.
Purpose: Scenarios are used by developers to obtain insight into the problem domain. On the other hand, scenarios are a good starting point for system visions and the finding of materials.
Relationships: Found materials are included in the glossary.

The production manager takes the *production orders* out of his in-box and computes the supplies of *raw materials*. After checking the *stock* he orders raw materials if the stock is not sufficient. If the stock is sufficient he creates the *work instructions* for a worker. After checking the occupation of the workers he sends the work instructions to a specific worker via his out-box.

Table 1: A scenario for operations scheduling of chemical production

System Vision

Description: System visions (see table 2) anticipate future work situations and are often supported by prototypes. We use several types of system visions in the development process, e.g. general visions, procedural visions, component visions.
Elements: A system vision consists of

- a name,
- a prose description of the anticipated use scenario,
- a context, in which the system vision takes place,
- probably a screen shot and
- one or more references to scenarios.

Participants: System visions are created by the developers. Prototypes often are based on system visions. While prototypes are created by the developers like system visions, prototypes in contrast are reviewed by the anticipated users.
Purpose: System visions are used to communicate within the community of developers about certain aspects of the future system. They are a starting point for prototypes and the future

system. Prototypes are used to communicate certain aspects of the future system with the anticipated users.
Relationships: System visions often are developed from previous written scenarios and will therefore contain references to these scenarios.

The production manager opens his in-box by double-clicking on the in-box icon. He sorts all documents in the in-box by type and selects all production orders. They are taken via drag and drop into the scheduling tool. The scheduling tool computes the supplies of raw materials which the production manager compares to the actual stock shown by the stock information tool. After ensuring the existence of sufficient raw material he creates a new work instruction document within the instruction editor. He completes the work instruction and drags it over to the out-box, where it is dropped. An addressee window appears, where the production manager fills in the name of the addressed worker. The name of an available worker is provided by the scheduling tool.

Table 2: A procedural system vision for operations scheduling

Glossary

Description: In the glossary, the relevant terms and materials of the domain (e.g., “production order” in the example in table 1) and the future system are described in a short and concise form. They are somewhat similar to domain specific dictionaries. A part of a sample glossary is shown in table 3.
Elements: The glossary consists of glossary entries. Each glossary entry has

- a name,
- a short description and
- a reference to other glossary entries, when appropriate.

Participants: The glossary is created and extended by the developers and used by all persons involved in the development process (individuals, representatives of the organization and developers). It is reviewed by users.
Purpose: We use the glossary to create a common language between developers and the domain experts. The glossary plays a central role in the whole development process, because all other document types used during the development process are linked through the central domain terms to it.
Relationships: There are no explicit references to other document types, while all other document types refer to the glossary.

• **work instruction**: all actions to be performed by a specific worker to assemble a product

Table 3: A part of a sample glossary

Cooperation Picture

Description: Cooperation pictures visualize the cooperation between organizational actors. There can be several kinds of organizational actors, including organizational units, whole organizations, functional roles and persons. Cooperation pictures are created for the crucial joint tasks in an organization. Ordinarily we create cooperation pictures for single joint tasks, but they can visualize several joint tasks at one time when appropriate.

We annotate the cooperations with a name and symbol for the information respectively material, which is transferred. We think that cooperation pictures have a considerable value for analyzing cooperations in an organization and organizational structures. We believe that they - in concert with ORC cards and location pictures - show exactly the information, that is needed for analyzing organizational changes.

Organizational actors in an cooperation picture can be refined through cooperation pictures when appropriate. A sample of such an extended cooperation picture is shown in figure 1.

Elements: A cooperation pictures consists of

- a name of the joint task,
- the involved organizational units (shown as icons or rectangles with annotated names)
- external organizational units and roles (shown as icons or clouds with names)
- cooperations between the organizational units (shown as arrows, starting at the organizational unit that initiates the cooperation)
- the media which is used for cooperation (shown as small icons annotated at the arrows) and
- the information or object used in cooperation (shown as text and icons annotating the arrows).

Participants: Cooperation pictures are prepared by developers based on interviews and previously created scenarios. They are discussed by developers and the domain experts, who are involved in the accomplishment of a given joint task. During this discussion, the cooperation pictures are modified by the developers.

Purpose: Cooperation pictures can be used from the very beginning of a project as an analysis document and evolve to a design document. Cooperation pictures are used by developers, future users and representatives of the organization to analyze the current cooperative work situation and possible enhancements. Moreover the developers can learn a lot about the specific cooperation tasks and coordinations needed to fulfill these tasks. When moving to the design of the future system cooperation pictures show the work tasks that are to be supported and provide a basis to anticipate and discuss organizational changes.

It is important to notice that we do not use cooperation pictures for the specification of the future system. The central purpose of cooperation pictures and the other described documents is to gain insight into the application domain which enables us to build adequate software.

Relationships: The places, organizational units and roles should be described in the glossary. This also holds true for the media, which are used for coordination purposes. Materials used in this context are central for the organization. Purpose tables describe the purpose of the cooperations shown in cooperation pictures. The scenarios should reflect these cooperations.

Cooperation pictures differ from most other notations for visualizing cooperations in organizations. Compared with action workflow loops (Medina-Mora, Winograd, Flores, Flores 1992) we see the following main differences:

- **Purpose**: Action workflow loops are used for specification purposes while we use cooperation pictures to gain insight into the work of the employees in an organization.
- **Notation**: In our approach, the organizational actors and the specific style of cooperations between them play a central role. Thus, every organizational actor is only shown at one place in the cooperation pictures, and the cooperations are annotated with the transferred information or material. Since action workflow loops focus on the conditions of satisfaction and the workflow in an organization, a business process map

shown with action workflow loops shows the conditions of satisfaction and the flow of information. In order to achieve this, the transferred information or materials are not shown and the participants are duplicated.

- **Applicability**: Since action workflow loops aim to specify the flow of information in an organization they fall short when we have to deal with informal work situations.

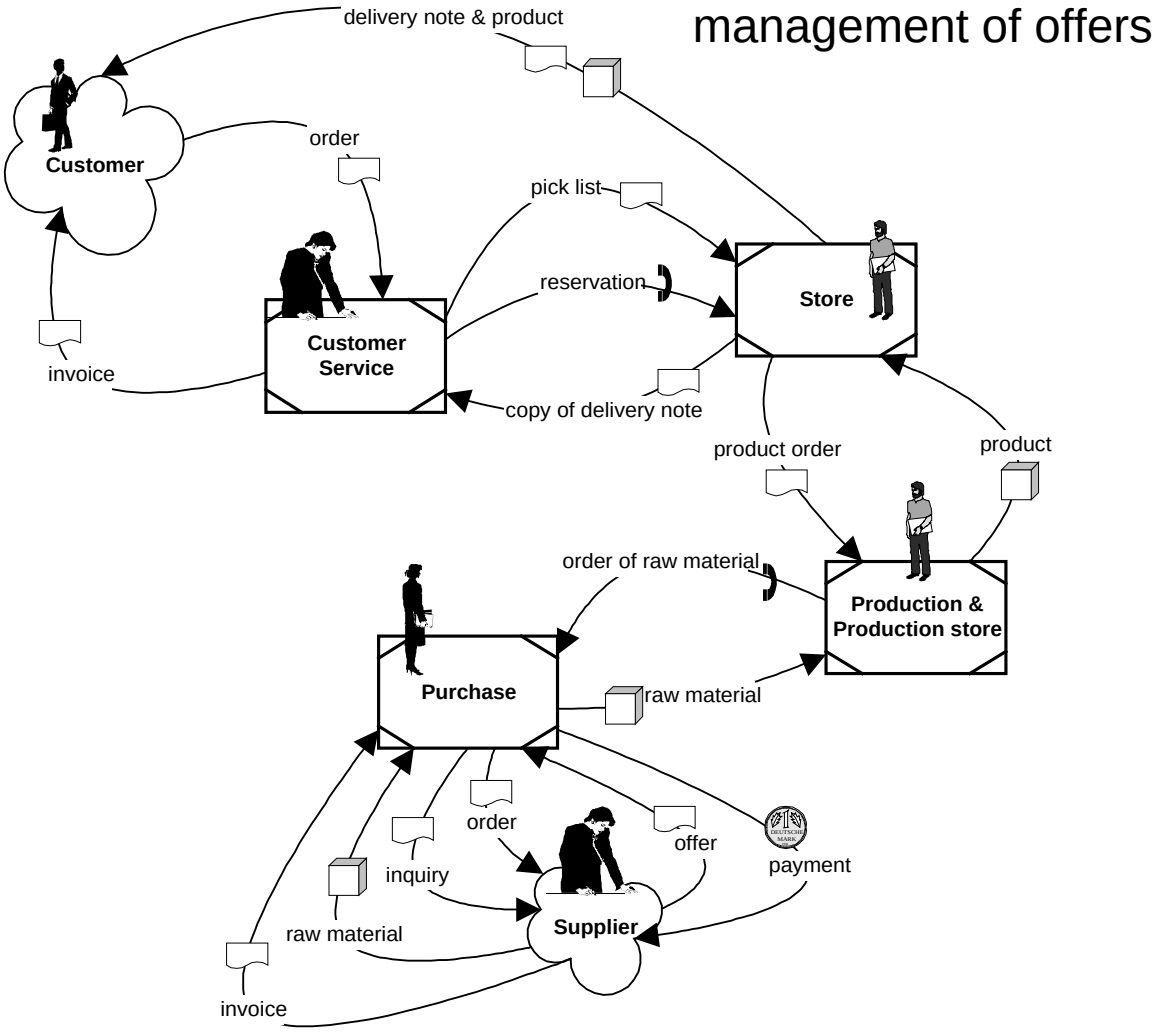

Fig. 1: A cooperation picture for management of offers

Purpose Table

Description: Purpose tables (an example is given in table 4) show why certain cooperative activities are performed.
Elements: A purpose table consists of the name of the performed task and a table with individual activities. Each entry in the table contains the following:

- a description of one or several single activities
- the purpose and implications of this single activity for coordination.

Participants: Purpose tables are created by the developers based on interviews.
Purpose: We use purpose tables as a supplement for cooperation pictures to analyze cooperations and possible enhancements and changes to the cooperations. Purpose tables are created with reference to cooperation pictures.

Relationships: The persons performing the tasks are located in the organizational units or roles of the cooperation pictures. The task described by the purpose table is a task that is performed by an organizational unit. The mentioned materials are found in the glossary.

Who does what, with what and whom	Why
A client (e.g. store) puts a production order in the in-box of the production manager.	The production manager can get the new production orders from his in-box.
The production manager gets a production order from his in-box, creates work instructions puts this plan into his out-box for work instructions.	A worker can get the work instructions from the out-box of the production manager.
A worker gets a work instruction out of the out-box of the production manager and produces the product. The product is brought to the requesting client.	Client can deposit new products.
The production manager fills out an order form for raw material and brings it to the purchaser.	Purchaser can buy the requested raw material.
Purchaser brings obtained raw material to the production unit.	Worker can deposit raw material and notify the production manager. The production manager can take the new raw material into account for future planning.

Table 4: A purpose table for processing a production order

ORC Card

Description: ORC cards (see figure 2) show for each organizational actor the responsibilities and the collaborations performed in fulfilling these responsibilities. They are based on CRC cards (CRC = class, responsibility, collaboration, see Wirfs-Brock, Wilkerson, Wiener 1990, original by Cunningham and Beck 1986) and are used in a similar way. They can be used to analyze and restructure the cooperations. The meaning of R and C is in ORC cards the same as in CRC cards. R stands for **R**esponsibility and C for **C**ollaboration. The O stands for **O**rganizational actions and can therefore have several meanings dependent on the viewed level of abstraction. It can stand for organizational member, organizational role, organizational unit or simply for organization.

We use ORC cards to describe the responsibilities and collaborations of the organizational actors. ORC cards can be used similar to CRC cards to gain a better understanding of the division of responsibilities between the participants and the dynamics of the collaborations. While stepping towards the design of the software system, ORC cards can be used to reason about changes in the organizational structure. They are suitable for thinking about changed responsibilities and collaborations in an organization. Changes in the organizational structure can be anticipated and ORC cards can be used for *organizational prototyping*.

Elements: An ORC card consists of

- the name of the person, role, organizational unit or organization,
- the fulfilled responsibilities and
- the collaborations that are needed to fulfill the responsibilities.

Participants: ORC cards are created by the anticipated users and representatives of the organization with the help of developers.
Purpose: Although - or because? - ORC cards do not describe the dynamic properties of cooperations they are suitable to gain a better understanding of the dynamics of cooperations. Because while checking the dynamics, we will frequently know the involved cooperation partners but not yet know their responsibilities. So, if there is no suitable responsibility we notice that we have missed something.

Additionally, ORC cards can be used to analyze the division of responsibilities between the organizational units. They can be changed to fulfill certain organizational requirements and then be used to simulate the central joint tasks of the organization (similar to the way we work with CRC cards). In this way, ORC cards provide a kind of consistency check for organizations and can evolve to a specification for the future organizational structures.
Relationships: The persons, roles, organizational units and organizations are represented in the ORC cards and cooperation pictures. While cooperation pictures show the cooperations between organizational units for *one joint task*, ORC cards show the *overall responsibilities* of a person, role, organizational unit or organization for *all joint tasks*. The responsibilities of the ORC cards map to the tasks, shown as arrows in the cooperation pictures. While these cooperations are not mentioned explicitly in cooperation pictures, they are in ORC cards. Thus, ORC cards externalize the knowledge of tasks which is known by the employees. The cooperations mentioned in the ORC cards map to the arrows between units in cooperation pictures.

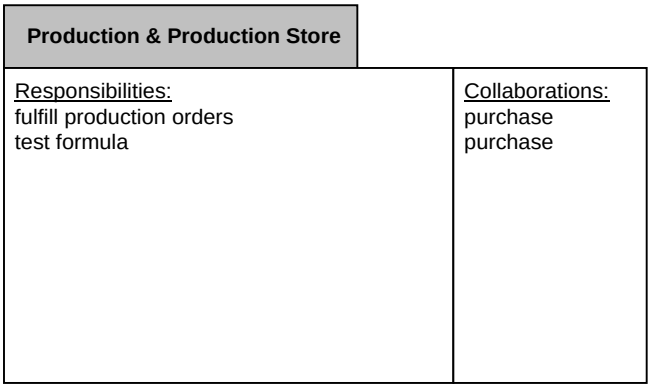

Fig. 2: An ORC card

Location Picture as a Special Kind of Cooperation Picture

Description: Location pictures show which locations are involved in what kind of cooperations. In contrast to „normal“ cooperation pictures they don’t show every cooperation. Instead of showing every single cooperation, location pictures show the *cooperation channels* between the organizational actors. A cooperation channel describes a means of cooperation, for example „electronic mail“ or „transfer of physical materials“. With this in mind we can see cooperation channels as abstracted cooperations. We will see, that there is a close relationship between cooperation channels and the cooperation media used in the future system. This enables a smooth transition from analysis to design.

Location pictures show which organizational actors and materials exist in the different places of the organization. While we visualize only one or few joint tasks in „normal“ cooperation pictures, we show all joint tasks for one unit in location pictures. Location pictures can and should be seen as a special kind of cooperation picture, since they show cooperations between „places“.

Figure 3 shows a location picture as a refinement for the organizational unit „Production & Production Store“ from the cooperation picture in figure 1.

Elements: A location picture consists of

- locations (shown as icons and names)
- roles (shown as icons and names in locations) and
- cooperation channels (shown as arrows with annotated icons and descriptions)

Participants: Location pictures are created by developers and are reviewed by the anticipated users.

Purpose: Location pictures are a starting point for the creation of the computer supported working environment for the future users. Location pictures first describe the current work situation of the future users and can evolve to a specification of the future system.

Relationships: The persons, roles, organizational units and organizations shown in location pictures are those from cooperation pictures and ORC cards. The cooperation channels between locations are an abstraction of the concrete cooperations shown in cooperation pictures and ORC cards. The materials shown in the locations are those, which are shown in cooperation pictures. Of course the materials are also mentioned in scenarios and described in the glossary. Location pictures have a strong relationship to system visions since both document types are used for system design. System visions show the potential software support for one - possibly cooperative – task, and location pictures show how persons and materials are embedded in locations and the organizational context.

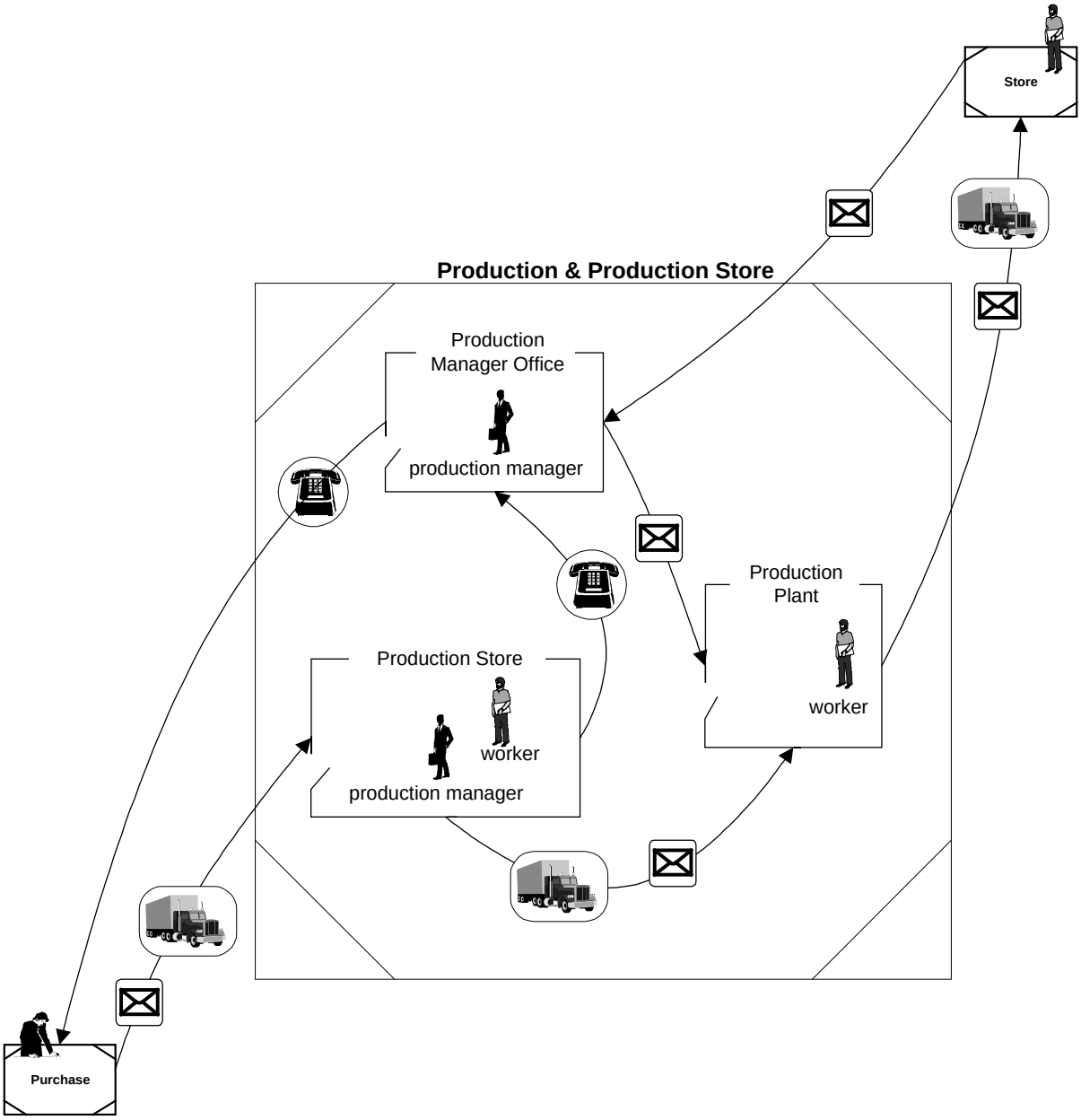

Fig. 3: A location picture for the organizational unit „Production & Production Store“

The development process with the described document types

We have introduced the well-known document types and the newer ones for cooperative work tasks and their relationships. But how do we use them in the development process? Let us look at a typical example: The development process for cooperative work tasks could start with interviews. Developers can write scenarios on the base of these interviews and get initial idea of the cooperations performed. Scenarios are then reviewed by the domain experts. When developers have gained a basic understanding of single work tasks, they could then assemble some cooperation pictures and write the first ORC cards. We think that the parallel development of cooperation pictures and ORC cards is very useful since both document types

focus on different aspects of cooperative work tasks. Because of these two views on cooperation, any errors and incompleteness can be found relatively easily. After developers have obtained an insight into single and joint work tasks they can create location pictures and initial system visions. The location pictures can be created based on the cooperation pictures and ORC cards. The organizational actors can be found in the cooperation pictures and ORC cards. When looking at the cooperations described in cooperation pictures and ORC cards we get first ideas about the cooperation channels between the organizational actors and places in the location picture. The organizational actors and places in the location pictures are a good starting point for reasoning about the workplaces for the future system. The cooperation channels can be transformed to cooperation media[3] used in the future system.

Because of our evolutionary approach, all development activities should be performed iteratively and with respect to the situation at hand. Therefore, cooperation pictures, ORC cards and location pictures usually will evolve during the development process from a description of actual work situations to a specification for the future software system. But a totally different project situation with other contextual constraints might call for a different usage of the above mentioned document types.

Location as an analysis and design metaphor

We have seen the central role of *persons, roles, organizational units* and *organizations* as *organizational actors* in the previous section since the document types for cooperative work are linked together by these. All of them have a concrete physical location, when cooperation takes place. Søregaard (see Søregaard 1988) has pointed out the central role of domain motivated location for cooperative work. He argues that the location of (shared) material plays an important role for coordination purposes. In our opinion this is also true for persons, who move to specific locations for cooperation purposes.

In systems built upon the desktop metaphor, the location of a person is always his or her individual workplace. In cooperative systems built upon the concept of location, persons can be at different locations at different times. To capture this conceptional extension we introduce the notion of working environment as an extension to the individual workplace.

The workplace is the location, where a person performs activities for individual or cooperative work tasks. The workplace is private to its owner and can’t be accessed by others. The owner of the workplace arranges tools and materials in a way that fits his or her needs best. A well known and suitable metaphor for concretion of the workplace is the desktop.

The working environment is the set of locations reachable by a person. Typically, it includes the workplace of this person. We can look at working environments either from the viewpoint of one individual or from the viewpoint of a group of individuals. In the latter case, the working environment for that group is the set of locations, which are reachable by this group. Working environments for groups are often designed with a specific set of cooperative work tasks in mind. In our opinion, a suitable concretion of working environments is the room metaphor. The room metaphor is described in a following section and a system vision visualizing the room metaphor is given at the end of this paper.

The room metaphor as a concretion of the location concept

We extend the tools and materials approach by the room metaphor. The room metaphor provides the user with the notion of software rooms. Software rooms share most of their

[3] We introduce some cooperation media in the next chapter.

characteristics with physical rooms. Rooms are connected by doors and users can move through rooms. Materials and tools are located in rooms and can be moved by the users to other rooms. Users and materials are exactly in one room at a given time. Cooperation can happen between and in rooms. Moreover, the software rooms form the spaces where we place cooperation media. Therefore, the room metaphor provides users and developers with a conceptional structure for the design of cooperative software systems.

The desktop metaphor can be used for the individual workplace which is located in the private room of an ermployee.

The room metaphor is similar to the places of Taligent's People, Places and Things metaphor (Cotter 1995). Places also represent the concept of location but in a more abstract way. We believe that users of a software system will intuitively know better how to deal with something concrete like rooms than something abstract like places.

Root (1988) also presents a system based upon rooms but has a different focus on *social browsing*. We do not intend to recreate social phenomena in software systems, because we don't think that software mediated communication can or should replace real communication. Otherwise social communication will lack efficiency if it is mediated by software. Our approach to the room metaphor focuses on coordination and cooperation.

Dourish and Harrison (1996) contrasted the terms *place* and *space*. While the abstract concept of *space* maps directly to our *locations*, our *rooms* are a concretion of *places*.

A Set of Cooperation Media for Supporting Cooperative Work Tasks

We have shown several documents for analyzing and designing cooperative work tasks. Additionally, we need appropriate *cooperation media* to shape the future system. In line with the tools and material approach we choose cooperation media with a strong similarity to the media used in the physical world. We will present some possible cooperation media for that purpose. The described cooperation media have been discussed with a rather technical focus in the literature in some detail. Nevertheless, our list is certainly not complete. It will be a future task to elaborate additional cooperation media for the design of cooperative systems. The presented design metaphors are described in the following pattern:

Description Pattern for Cooperation Media
Description: What does the cooperation medium look like?
Purpose: What cooperative situation(s) do we use the cooperation medium for?

Office Boxes

Description: Office boxes link locations together and allow persons at these locations to exchange materials. Materials can be put into out-boxes and are transferred by an automation into the in-box of the addressed person. Office boxes provide especially the possibility to transfer materials between private workplaces of individuals. Office boxes were introduced by Wulf, Gryczan, Züllighoven (1996) together with the notion of process pattern (see next section).
Purpose: Office boxes can be used for asynchronous and unidirectional cooperation through the transferred material.

Process Pattern

Description: Process patterns (see figure 5) describe how individuals participate in a asynchronous cooperative work task. They emerge, when people engaged in the work process feel the need for articulating their cooperative work. Process patterns show which tasks on

which materials are performed by whom (person or role) in which order, but do not determine, how these tasks are to be performed. In contrast to the context of workflow management systems, process patterns are materials for the users of the system and can be manipulated by them to suit their needs. Therefore, the users are still able to change their working style. Materials can be put into a portfolio on which a process pattern can be attached. Portfolios can be put into out-boxes, from where they are sent to the office box of the addressee. Process patterns are introduced by Wulf, Gryczan and Züllighoven (1996).
Purpose: Process patterns describe and - in concert with office boxes - support a sequence of asynchronous cooperations for one joint task.

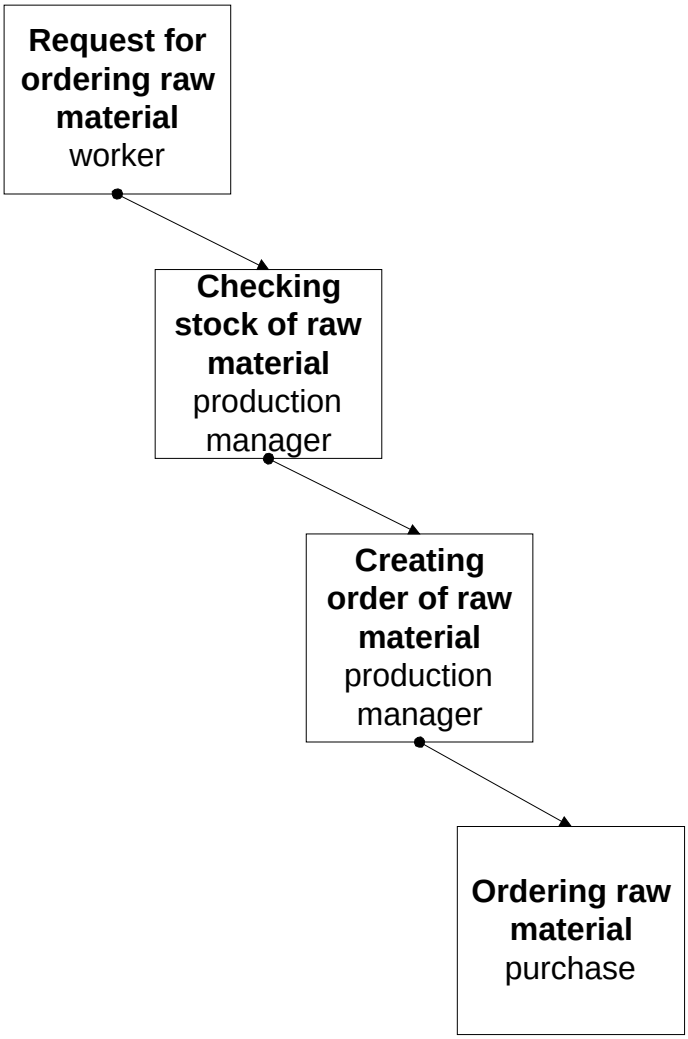

Fig. 5: A process pattern for ordering raw material

Bulletin Board

Description: Bulletin boards are placed at public locations and are used for information announcements between a group of persons. A feature specific to bulletin boards is that they can't be moved. Therefore, people find specific bulletin boards everytime at the same place.

Examples of bulletin board systems for supporting cooperation are discussed in the literature, for example Sarin, Greif (1985).
Purpose: Bulletin boards can be used for asynchronous cooperation. In contrast to office boxes there is a relatively unspecified set of addressees.

Conference Table

Description: Conference tables are used by a set of persons for real time cooperation and coordination. The persons "sit" around the table and work on shared materials. Characteristic for that kind of cooperation is the autonomous coordination of the participants working together on one shared material at a given time.

Mantei (1988) describes some aspects of electronic meeting systems based upon physical conference tables. In many systems for supporting synchonous cooperations in small groups the conference table could be quite adaquate to explain, where the action takes place, although this is not mentioned explicitly in most cases. Examples of such systems are presented by Newman-Wolfe, Webb, Montes (1992), Tang, Leifer (1988), Ellis, Gibbs, Rein (1988) and Sarin, Greif (1985).

Purpose: Conference tables are used for synchronous cooperation at one location.

A system vision for a joint task supported by the room metaphor

In this section we present a sample system vision which shows the use of the room metaphor in a cooperative software system. The used cooperation media and the terms related to the room metaphor are written in *italics*.

The production manager takes the production orders out of his *in-box* and computes the supplies of raw material with the help of the scheduling tool. He then double-clicks on the *door* icon, representing the production store. A window for that *room* appears and shows that somebody is already in the production store. He opens the stock tool in the production store and sees that another person is working on the stock. He notices that the stock of the required raw material is not sufficient and clicks on the people-icon at the window border to get a list of the people currently working on the stock. He clicks on the name of the only present worker and a *chat* window is opened. Through the chat with the worker, he finds out that the worker currently is depositing the needed raw material. The production manager leaves the production store by closing the window of that *room*. Now he creates the work instructions with the gained information in mind. After the work instructions are created and an available worker is assigned, the production manager sends the work instruction to that worker via his *out-box*.

In figure 6 and 7 two screen shots from the WIGWAM[4] prototype are shown. In figure 6 the individual workplace room is shown on the left hand side. In this room there are two materials, shown as icons. At the lower bound of this room there are several icons which we find in every room: tools, chat, in-box, out-box, room plan, persons, help, exit. The tool icon starts the toolbox where one can choose a tool and start it. The chat icon starts up a chat tool, with wich the user can „talk“ via the keyboard with other users. The in-box and out-box icons show the in-box respectively out-box of this room. The room plan icon opens the room plan where all the rooms in the system are displayed (figure 7 shows a room plan). The persons icon opens a window with all the persons currently beeing in the room or looking into the room. The help icon starts a web browser with a help page for the room and the exit icon closes the room. At the right hand side of the window there are two additional rooms. The user *is in* the workplace room and *looks into* the two other rooms. With a click in one room, the user enters this room. Materials can be moved via drag and drop from one room to another.

[4] WIGWAM is the acronym for „**W**orking **i**n **G**roups“ with **WAM**. WAM is an acronym again and stands for the german translation of the Tools & Material metaphor „**W**erkzeug, **A**utomat, **M**aterial“.

Figure 7 shows the room plan in which all the rooms in the system are displayed. The user can choose one room by double-clicking. After choosing a room the users enters the room.

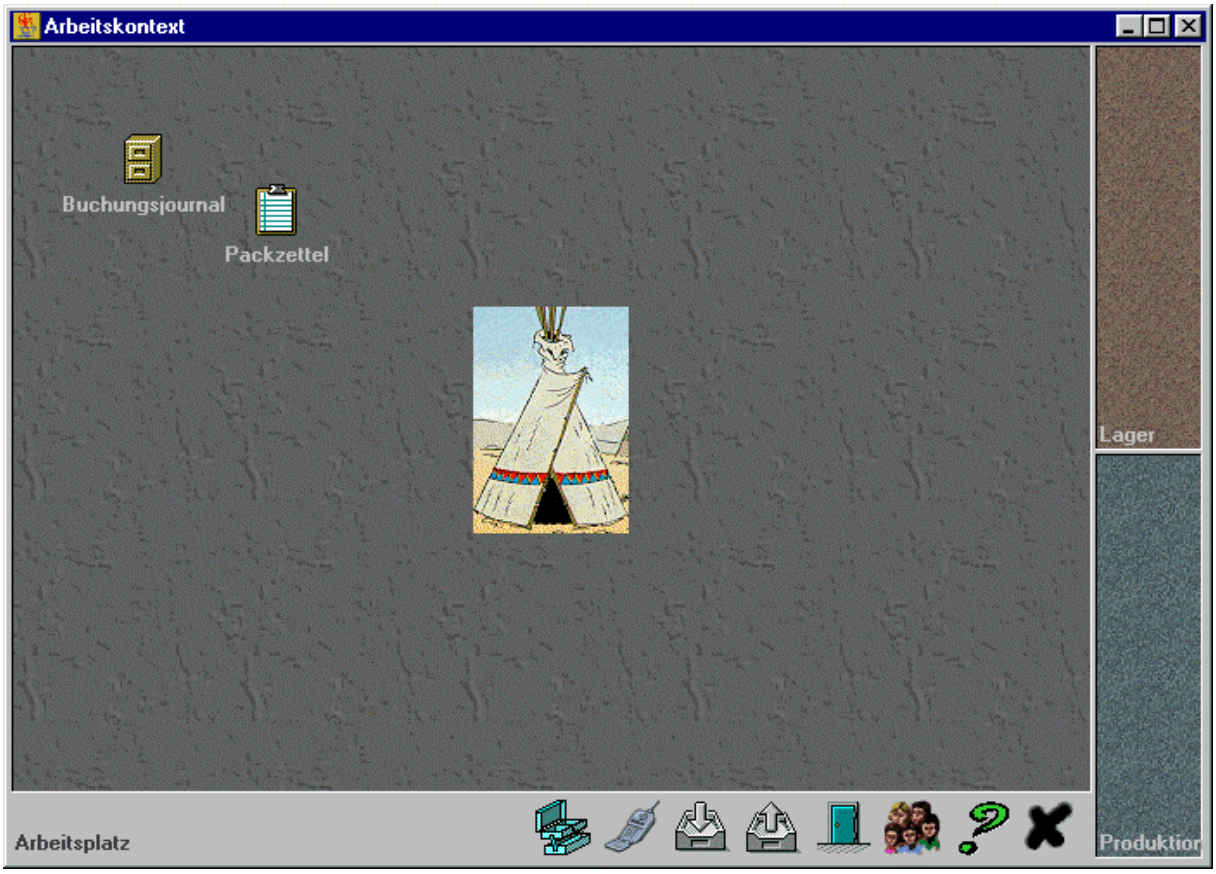

Fig. 6: The workplace and two rooms currently in view.

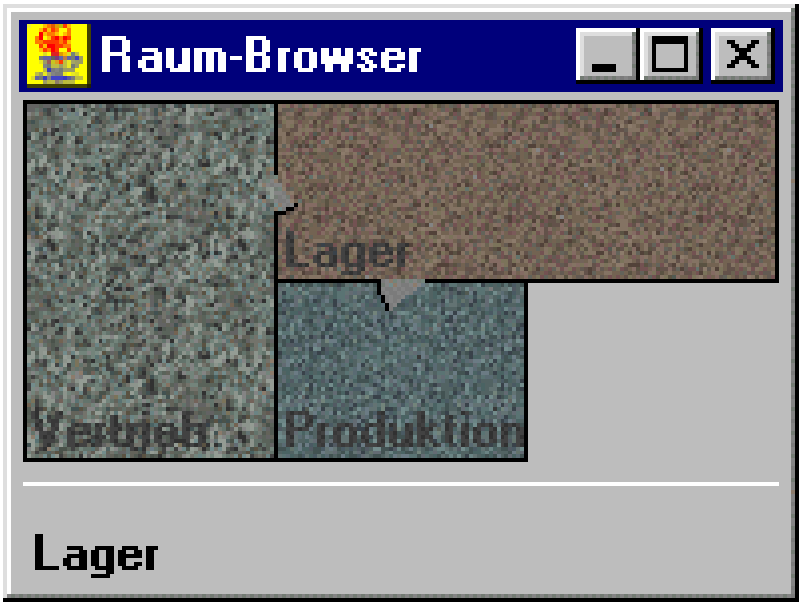

Fig. 7: The rooms in the system shown by the Room-Browser

Summary

We have presented our approach to the analysis and design of cooperative work tasks, which is based upon the tools and materials metaphor. The development process is supported by several document types which we have presented here. We have described cooperation pictures and introduced ORC cards and location pictures. We have shown that these new document types, in concert with the well-known document types, provide the information needed for the design of cooperative systems.

We have argued that these document types should be supplemented by several cooperation media for cooperative systems. Some cooperation media for that purpose have been presented, but they have to be supplemented by additional cooperation media. That will be one of our future tasks.

The described document types in concert with the evolutionary development process, the room metaphor and the cooperation media provide the mechanisms for a rich and adequate requirement analysis and the smooth transition from analysis through design to an adequate software system supporting cooperative work tasks.

References

Dirk Bäumer, Rolf Knoll, Guido Gryczan, Heinz Züllighoven (1996): *Large Scale Object-Oriented Software Development in a Banking Environment - An Experience Report*. In: Pierre Cointe (Ed.): ECOOP'96 - Object-Oriented Programming, 10th European Conference, Proceedings, Springer Verlag, Linz, Austria, July 1996, pp. 73-90.

Sean Cotter with Mike Potel (1995): *Inside Taligent Technology,* Reading, MA: Addison-Wesley 1995.

Ward Cunningham, Kent Beck (1986): *A Diagram for Object-Oriented Programs.* In: SIGPLAN Notices. Volume 21(11). November 1986.

Paul Dourish, Steve Harrison (1996): *Re-Place-ing Space: The Roles of Place and Space in Collaborative Systems.* In CSCW 96: Proceedings of the Conference on Computer Supported Cooperative Work, Bostan, Massachusetts, 1996.

C.A. Ellis, S.J. Gibbs, G.L. Rein (1991): *Groupware. Some Issues And Experiences*. In Communications of the ACM, Vol. 34, Number 1, January 1991, pp. 38-58.

Christiane Floyd (1992): *Software Development as Reality Construction*. In: Christiane Floyd, Heinz Züllighoven, Reinhard Budde, Reinhard Keil-Slawik (Eds.): Software Development and Reality Construction, Berlin 1992, pp. 86-100.

Anita Krabbel, Sabine Ratuski, Ingrid Wetzel (1996): *Requirements Analysis of Joint Tasks in Hospitals*. In: B. Dahlbom at al. (Eds.): IRIS 19 „The Future", Proceedings of the 19th Information system Research seminar in Scandinavia, 10 - 13 August 1996 at Lökeberg, Sweden. Gothenburg Studies in Informatics, Report 8, June 1996, pp. 733-749.

Anita Krabbel, Ingrid Wetzel, Sabine Ratuski (1996): *Participation of Heterogeneous User Groups: Providing an Integrated Hospital Information System*. In: J. Blomberg, F. Kensing, E. Dykstra-Erickson (Eds.): PDC'96 Proceedings of the Participatory Design Conference, Cambridge, Massachusetts, November 1996, pp. 241-249.

Meir M. Lehmann (1980): *Programs, Life Cycles, and Laws of Software Evolution*. In: Proceedings of the IEEE. Volume 68, Number 9. September 1980.

Marilyn Mantei (1988): *Capturing the Capture Lab Concepts: A Case Study in the Design of Computer Supported Meeting Environments*. In CSCW 88 Proceedings of the Conference on Computer Supported Cooperative Work, Oregon, 1988.

Raúl Medina-Mora, Terry Winograd, Rodrigo Flores, Fernando Flores (1992): *The Action Workflow Approach to Workflow Management Technology*. In: CSCW'92 Proceedings of the Conference on Computer Supported Cooperative Work, Toronto, Canada. 1992, pp. 281 - 288.

R.E. Newman-Wolfe, M.L. Webb, M. Montes (1992). *Implicit Locking in the Ensemble Concurrent Object-Oriented Graphics Editor*. In CSCW'92 Proceedings of the Conference on Computer Supported Cooperative Work, Toronto, Kanada. 1992.

Dirk Riehle, Heinz Züllighoven (1995): *A Pattern Language for Tool Construction and Integration Based on the Tools and Material Metaphor*. In: James O. Coplien and Douglas C. Schmidt (Eds.): Pattern Languages of Program Design. Reading, Massachusetts: Addison-Wesley.

Robert W. Root (1988): *Design of a Multi-Media Vehicle for Social Browsing.* In CSCW 88 Proceedings of the Conference on Computer Supported Cooperative Work, Oregon, 1988.

Sunil Sarin, Irene Greif (1985): *Computer-based real-time conferences.* Laboratory for Computer science at Massachusetts Institute of Technology. Juli 1985.

Pål Sørgaard (1988): *Object Oriented Programming and Computerised Shared Material.* In ECOOP'88: European Conference on Object-Oriented Programming. Proceedings. Springer Verlag 1988.

John C. Tang, Larry J. Leifer (1988): *A framework for understanding the workspace activity of design teams*. In CSCW'88 Proceedings of the Conference on Computer Supported Cooperative Work, Oregon, 1988.

R.J. Wirfs-Brock, B. Wilkerson, L. Wiener (1990): *Designing Object-Oriented Software*, Englewood Cliffs, NJ: Prentice Hall, 1990.

Martina Wulf, Guido Gryczan, Heinz Züllighoven (1996): *Process Patterns - Supporting Cooperative Work in the Tools & Materials Approach.* In: B. Dahlbom at al. (Eds.): IRIS 19 „The Future", Proceedings of the 19th Information system Research seminar in Scandinavia, 10 - 13 August 1996 at Lökeberg, Sweden. Gothenburg Studies in Informatics, Report 8, June 1996, pp. 445-460.