

Herrmann, Thomas; Hoffmann, Marcel; Jahnke, Isa; Kienle, Andrea; Kunau, Gabriele; Loser, Kai-Uwe; Menold, Natalja (2003): Concepts for Usable Patterns of Groupware Applications. In: Mark Pendergast, Kjeld Schmidt, Carla Simone, Marilyn Tremaine (Eds.): Proceedings of the 2003 International ACM SIGGROUP Conference on Supporting Group Work. . S. 349-358.

# Concepts for Usable Patterns of Groupware Applications

Thomas Herrmann, Marcel Hoffmann, Isa Jahnke, Andrea Kienle, Gabriele Kunau,  
Kai-Uwe Loser, Natalja Menold

Informatics & Society, University of Dortmund, Germany  
44221 Dortmund, Germany

+49 231 755 2779

{thomas.herrmann, marcel.hoffmann, isa.jahnke, andrea.kienle, gabriele.kunau, kai-uwe.loser,  
natalja.menold}@udo.edu

## ABSTRACT

Patterns, which are based on in-depth practical experience, can be instructing for the design of groupware applications as socio-technical systems. On the basis of a summary of the concept of patterns - as elaborated by the architect Christopher Alexander - its adoptions within computer science are retraced and relationships to the area of groupware are described. General principles for patterns within this domain are formulated and supported by examples from a wide range of experience with knowledge management systems. The analysis reveals that every pattern of a groupware application has to combine the description of social as well as technical structures, and that a single pattern can only be understood in the context of a pattern language. It also shows that such a language has to integrate patterns of socio-technical solutions with measures and procedures for introducing them, and that the language not only has to express one type of directed relationship between the patterns but a variety of different types which have to be deliberately assigned to the patterns.

## Categories and Subject Descriptors

H.5.3 [Information Interfaces and Presentation]: Group and Organization Interfaces – *computer-supported cooperative work, organizational design, theory and models.*

H.4.1 [Information Systems Application]: Office Automation – *groupware.*

K.4.3 [Computers and Society]: Organizational Impacts – *Computer-supported collaborative work.*

## General Terms

Documentation, Design, Human Factors, Management.

## Keywords

Pattern, Pattern language, CSCW, Knowledge management, socio-technical systems.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

GROUP'03, November 9–12, 2003, Sanibel Island, Florida, USA.

Copyright 2003 ACM 1-58113-693-5/03/0011...\$5.00.

## 1. INTRODUCTION

### 1.1 Relevance of patterns

With increasing experience, groupware researchers and practitioners are able to develop ideas for solutions for certain constellations where groupware is applied. These ideas are usually backed by solutions which have been successfully implemented in real world settings – and can therefore be considered as a kind of pattern. They not only refer to technical features but also to the interplay between the technical system, the users, their tasks and organizational constellations. We call this complex interplay “socio-technical system” [23]. Descriptions of groupware solutions as socio-technical systems are difficult to formalize, as this type of system is very context-sensitive due to its dependency on human actors and organizational structures [35]. Although every situation of a groupware application has its specific characteristics, researchers and practitioners usually try to start with concepts or solutions they are already familiar with. The next step is to then adapt this solution to the current case. For an expert it is of course, reasonable to rely upon proven solutions when designing a new application. Nevertheless, this approach can cause problems when experts fail to share their expertise with other parties involved in the case. Therefore existing ideas about appropriate solutions for a socio-technical system ought to be made explicit as far as possible, by using specific kinds of documentation.

The advantages of such documentation are manifold;

- They could help the decision makers and end-users of the system to discuss the features of the socio-technical system in advance in the course of a participatory design process.
- They could help users to benefit from the experience of others and to therefore avoid mistakes.

We assume that this approach of using patterns and a pattern language is a promising means of meeting these requirements. However, the problems involved in using such documentation are manifold, too. They mainly relate to the question of the appropriate kind of documentation for a socio-technical solution. Prototypes focus too much on the technical aspects, so called “reference models” [29] stick to a business process perspective. Other methods, such as UML, have a different purpose as they are developed to support the process of software development. The documentation of socio-technical solutions should combine formal specifications with explanations, and make it possible to integrate them into an entirety of documented solutions for varying situations, even if they are incomplete.

## 1.2 Origins of patterns

*„Each pattern describes a problem which occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing it the same way twice.“ ([3], p. x)*

This quote reveals much about patterns as defined and used by the architect Christopher Alexander and his colleagues back in the late 1970s [3], [4]. Alexander assumes that those places in our urban environment where people like to be, where they can enjoy themselves or successfully do their work, have characteristic traits that can be captured and described and then reused to build more places like them. The method of describing these characteristic traits is the usage of patterns. For the sake of clarity, comparability and readability Alexander created a format to articulate patterns ([3], pp. x).

Each Alexanderian pattern is composed of eight elements:

- |  |  |
|--|--|
| 1. Title   | 5. Problem – body                                      |
| 2. Picture   | 6. Solution  |
| 3. Introduction with references to larger patterns | 7. Diagram   |
| 4. Problem – headline                              | 8. Final paragraph with references to smaller patterns |

Although all patterns are written the same way, they can address quite different levels of urban design. There are global patterns such as “The distribution of towns” within a region and there are very local patterns such as “Small meeting rooms” within public buildings or “Built-in seats” within rooms [3]. All of these patterns are interrelated and together they form a pattern language describing the whole realm of urban architecture. Each pattern relates to larger ones which it helps to complete; and in turn each pattern refers to smaller ones which help to complete itself ([3], p. xii).

## 1.3 Usage of patterns within computer science

The idea of capturing knowledge and putting it into patterns has been adopted in various fields of computer science. Gamma, Helm, Johnson and Vlissides [13] transferred the concept of patterns in urban architecture to software engineering and published a widely received collection of design patterns for object-oriented software engineering. Jan Borchers [6] is a prominent exponent of the usage of patterns within the field of HCI. He offers good examples of design of user interfaces using patterns. Coplien’s [25] domain describes patterns of organizational processes in software engineering companies rather than the results of the design process. Thomas Erickson [12] being the one who brings in the term “lingua franca” for patterns stresses the fact that patterns can be a means of communication in interdisciplinary design teams, and therefore support participatory design processes. John C. Thomas works on a socio-technical pattern language to enable knowledge of the social sciences to be used in technical design processes [36] and to be put in the context of CSCW (e.g. CSCW2002 workshop).

Martin et al. [22] pick up a different aspect of Alexander’s work: Their goal is to use patterns as a way “through which the important findings of different studies are highlighted and presented in a manner that is more accessible to the CSCW community at large.” ([22] p. 43). While their work is very similar to Alexander’s ideas regarding the importance of real-world-observations, they move away from the idea of design patterns as a means of

providing solutions to given problems. In their attempt to use Alexander’s notion of patterns for their purpose the authors came to the conclusion that the “expressive power was limited” ([22] p. 45). One argument is that not all patterns found in material derived from field work are relevant. It is therefore not possible to present all patterns in the Alexanderian problem-oriented form. Our argument is that patterns were not meant to be a mere analytical tool. The power of patterns lies in their design-oriented approach, their main purpose being in helping to solve reoccurring problems. An abstract and discernible articulation of a common problem facilitates the pattern’s reuse and is therefore indispensable.

## 1.4 From Patterns to Pattern Languages

As we have explained, pattern languages in the Alexanderian sense are networks of interrelated patterns. This connotation has been widely adopted in computer science: pattern languages are usually collections of patterns belonging to the same domain. (Note: Martin et al [22] use the term differently. With pattern language they mean the formalism used to describe patterns). The internet has become a popular place to publish patterns; associated e-mail lists are often used to discuss pattern-related issues. Some web-sites just present patterns; others invite readers to contribute their own patterns in order to create a rich pattern language. An example of the latter kind is the public sphere project directed by Doug Schuler. As part of the project he publicly collects patterns for “living communication” and offers a platform for the discussion of patterns [31]. Other sites present patterns in the field of organization [25], cooperative interaction [20], groupware [14], HCI [15] or UI [37].

An open issue which we will address later on in this paper is how relationships between patterns can be acquired. Some authors such as Martin et al. [22] “feel that these should emerge once a number of patterns have been developed and put to use.” We think that much work has to be done to create a true pattern language from interrelated patterns and that profound knowledge of the domain is necessary to perform this work. This effort has to be put in if the patterns are to be of use to readers. Alexander describes very accurately how each project can derive its own individual pattern language from a given one, if a meaningful path has been laid through the network of patterns ([3], pp. xxxvii).

So far, we have described a variety of cases where patterns can be useful in the context of computer science. The next section describes the special characteristics of patterns for groupware applications. Since it is impossible to cover all possible kinds of patterns in this field, section 3 focuses on an exemplary area: knowledge management which includes most of the typical functionality and organizational structures of groupware applications. All authors of this paper have been doing research in the field of knowledge management for about four years and gathered manifold experience while introducing, developing and evaluating knowledge management solutions and working as consultants. The patterns of knowledge management presented in this paper are consolidated results of a qualitative study in nine German companies in various branches [18] [19] and a second research project which actively designed and introduced knowledge management solutions [9] [10]. The approach to knowledge management exemplified in our projects heavily relies on collaboration and the application of software for cooperative work and communication. Although there are approaches that do emphasize other

aspects of knowledge management, like knowledge representation or the measuring of knowledge capital, collaboration and communication are of great significance to the success of knowledge management from any perspective. Therefore it is hardly surprising knowledge management became one of the most favored areas for the application of groupware. Structures, processes and features which occurred repeatedly in our case studies formed the basis for our definition of patterns. Accordingly, the development of the patterns is considered as a meta-analysis of the results of our research projects. In this paper we give an overview of six different knowledge management patterns (subsection 3.1).

In subsection 3.2 we discuss various problems and aspects which have to be taken into account when the details of a pattern are specified. Since a socio-technical solution is inseparably interwoven with its process of introduction, we additionally give an example of a sub-pattern of this aspect (subsection 3.3). In subsection 3.4 we discuss possibilities of building a pattern language which bases on these knowledge management patterns. Section 4 deals with the discussion of benefits and risks of patterns for groupware applications and summarizes concepts for these patterns.

## 2. CHARACTERISTICS AND POTENTIAL USAGE OF PATTERNS FOR GROUPWARE APPLICATIONS

The functionality and purpose of a software system such as groupware can only be understood and designed in respect of the context where it is applied. The usage of a certain software system varies with the characteristics of the organization or the group which is employing it. These characteristics are related to the way of cooperative task completion, to the resources which are available, to the coordination mechanisms and conventions of an organization, to the willingness of its members to adopt new technology etc. Thus, we strongly argue that a pattern, which represents the solution of a real problem in terms of Alexander [3] has to consider not only technical aspects, but also the characteristics of organizational structures, conventions or roles which are involved. Groupware patterns should be application oriented and therefore help to understand and design socio-technical systems [11]. Typical examples are the support of business processes, of knowledge management, of collaborative learning, of cooperative decision making, or of joint design, programming or editing.

Obviously, a groupware application pattern includes a kind of model of the characteristics of cooperative work. Thus, the question arises, as to whether patterns are not a matter of the broad discourse of the limitations and potentials of task and work modeling or representation [5], [7], [28], [30], [34], [35]. The discussion about the categorization of human activities, the degree of appropriate formalization of work procedures, and the scope between maps and scripts seems to be relevant in this context and was reconstructed in respect of the impact of modeling methods in [17].

We want to emphasize that patterns are more than generalized and strictly defined procedures: they involve the experience and even wisdom of practitioners and researchers and can serve as a resource for the enhancement of human activities; thus making groupware application patterns more similar to maps in terms of Suchman [33] and Schmidt [30] than scripts.

A pattern of a groupware application should describe the structure (static aspect) as well as the potential processes (dynamic aspect). It should clearly differentiate between the activities carried out by people in certain roles and state transitions which are produced by technical components. However, both aspects – social and technical structure – should be co-presented in each pattern. Since social interactions are often informal, the description of groupware patterns must respect the potential of freedom of decision as well as uncertainty. It has to be emphasized that groupware applications include prescriptions of the appropriate conventions of human behavior. However, patterns of socio-technical systems have to be vague and incomplete instead of over-determining human behavior. This intentional incompleteness should not be confused with situations where we are not able to give a sufficient description of a solution. To avoid such confusion, a pattern should provide symbols to differentiate between these types of incompleteness. Furthermore, it should support varying degrees of explicitness by referring to more detailed sub-patterns or by offering links to graphical or pictorial representations.

As described above Alexander's template for patterns provides room for at least two illustrations: first there should be a picture "which shows an archetypal example of that pattern." ([3], p. x). Additionally there should be a diagram "which shows the solution ... with labels to indicate its main components". ([3], p. xi). For architectural patterns this seems to be more natural than for more abstract topics such as socio-technical systems design or cooperation support. However we argue strongly for the usage of illustrations in both places.

In urban architecture, photos help to support the aesthetical imagination for the future layout of an area or building; thereby they help the reader to decide whether or not a given pattern is appropriate for the problem at hand. The domain of groupware applications calls for more dynamic illustrations: a series of photos, a series of screenshots or short video clips may be employed to reveal a typical situation in which a system is used and thereby relate a technical system to the social context of its usage. As patterns are often published using electronic means such as the internet or CDs, multi-media is an appropriate way of including such documents.

The diagram summarizing the solution is also important in the domain of groupware; their purpose is not only to offer an appropriate description of the solution. There are more reasons as to why we suggest putting effort into finding graphical representations for the domain of groupware applications. Compared with exclusively text based representations graphical illustrations exhibit some clear advantages: They are easier to remember and recognize, they are helpful in drawing attention to certain parts of the representation, e.g. by marking points of reference in the course of communication, and they allow multiple alternative sequences of reading. As a consequence they can be used to support the testing of a designed solution and its effects using methods similar to cognitive walkthroughs. Due to these characteristics, diagrammatically represented groupware patterns can be a starting point for requirements elicitation and engineering and can facilitate processes of participatory design as shown in [8]. For these purposes, however, representations are needed that provide a sufficiently detailed description of a solution while, at the same time, leaving enough flexibility for human behavior. We could even imagine that – in a couple of years' time – commercial-off-the-shelf groupware could be delivered together with some pat-

terns of most successful application solutions. When considering its relationship to participative design it is important to acknowledge that the behavior of actors in socio-technical systems cannot be determined by any pattern's representation. Thus, the socio-technical patterns can only be resources that are used by actors in the process of their self-organization. Users of patterns should never think that they have a global formula for success at hand, but should be aware and able to see the special characteristics within every new setting.

### 3. PATTERNS OF KNOWLEDGE MANAGEMENT

The patterns described in this section are to a large extent based on two research projects. The first was a qualitative study of knowledge management initiatives we conducted in 2001/2002 [18] [19]. The main task here was to identify success-factors and obstacles that influence the introduction and usage of knowledge management systems and to derive technical and organizational design recommendations. The data was gathered by using observations and analyzing documents with the main source being semi-structured interviews, which were accompanied by discussion with those employees of the companies who were in charge of the technical and organizational aspects of the knowledge management solution.

The second research project which influenced the development of the patterns dealt with the design and introduction of knowledge management solutions in a consultant company and a public organization for consumer counseling. In both companies the commercial-off-the shelf software Livelink® was chosen. The aim of the research project was to develop innovative methods for the configuration and introduction of commercial knowledge management software for particular needs. Experiences gathered in this project cover all stages from early project establishment through to introduction and regular usage of the software [10] and include the evaluation of approaches to integrate knowledge management with industrial business processes [9].

Because of the wide range of different companies (different branches, sizes, employed technical solutions and in different phases of the introduction-project) we extract underlying principles and then generalize them into two types of patterns. The usage processes and corresponding features of knowledge management solutions found in various companies were defined as a pattern.

The escalating help-system – which is an example which explains the aspects of constructing a pattern (subsection 3.2) – was mainly

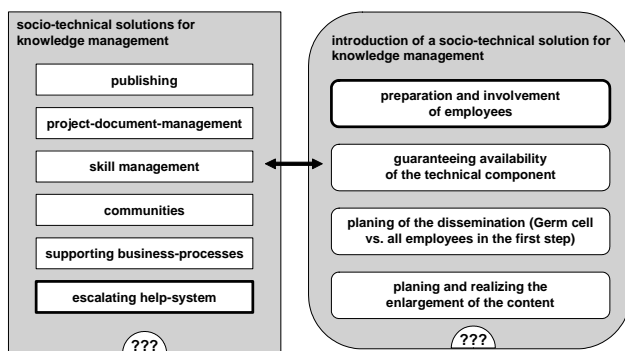


Figure 1: overview of knowledge management patterns

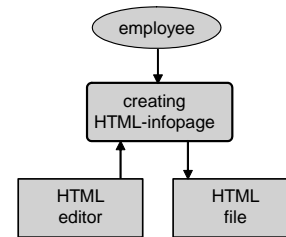


Figure 2: Simple example of the SeeMe notation

influenced by one of our nine studies that took place in an industrial company with more than 6.000 employees. The system was introduced at about 2.400 workplaces. The introduction started with a 2 hour training session for the employees. 14 days after the training session we conducted in-depth interviews with 16 randomly chosen users on different hierarchical levels, and of different educational backgrounds. Six weeks later, we analyzed their participation in the discussions to get an impression of how often employees contributed to the system. We repeated the monitoring of the participation in the system based discussions after the employees had used the system for 6 months. At this point we also took a closer look at the content type and structure of contributions. We used the results of this monitoring as a basis for questions in the follow-up interviews with three participants who had not used the system intensively and three who had. In this second round of interviews we paid special attention to the frequency of their system use and the reasons for it, as well as their views on the participation of other employees. We also asked for their ideas on how to improve the participation by the employees.

Figure 1 gives an overview of the different patterns we found. We combine patterns of socio-technical solutions and organizational patterns that address the introduction of these solutions.

For the diagram, we use a consistent modeling notation for semi-structured, socio-technical models – SeeMe [16] – which allows beginners to understand the diagrams very quickly but also offers symbols for more complex constructions. The simple example of fig. 2 shows the three basic elements: role (employee), activity (creating ...), and entity (HTML-Editor, HTML-File). While roles are related to the social aspects of a solution, entities can represent all kinds of resources – including technical systems. The direction of the arcs specifies their meaning: The editor is used by the activity which produces the HTML-File. The role carries out the activity. An arc which points from an activity to a role indicates that the role is influenced. Hexagons contain conditions, empty hexagons symbolize that the involved roles can specify the condition. Empty semi-circles specify incompleteness while question marks are used to express uncertainty.

From our point of view, the two types of patterns in fig. 1 are inseparably interwoven and may have a cyclic relationship, since they support and form each other reciprocally. In the following we describe an overview of the patterns of knowledge management and discuss the problems of constructing a pattern by referring to the “Escalating Help-System”. To give an impression of an introduction pattern, we roughly describe the pattern “preparation and involvement of employees”. This rough description is also used to demonstrate what a pattern looks like in accordance with Alexander’s scheme. An overview of all found patterns is impossible in this paper due to limited space. The patterns we describe in detail are marked with bold lines in figure 1. For all socio-

technical solutions (patterns at the left side in fig. 1) all organizational patterns (right side) are relevant and vice versa. For the sake of clarity this is shown by the arrow in the middle of the diagram.

### 3.1 Patterns of a socio-technical solution

During our practical work in the field of knowledge management we learnt that more than one perspective on this type of socio-technical system was relevant. On the one hand, we were inspired by the literature identifying a knowledge life cycle including phases such as knowledge identification, knowledge acquisition and explication, knowledge development and structuring, knowledge distribution, usage of knowledge and evaluation of knowledge (cp. [27]). These steps or phases are related to tasks of knowledge work which is conducted by employees who have to accompany their original work by using continuous knowledge improvement. If this improvement is to take place cooperatively, knowledge exchange is required. Usually, knowledge management and the underlying cycle are supported by technical components but success is mainly based on organizational efforts. Thus, the organizational and technical measures of the described knowledge life cycle can be considered as a global knowledge management pattern. However, each of the different phases listed do not provide a pattern in themselves as they do not solve a complete problem.

On the other hand we found that many companies have a specific problem or interest if they decide to employ knowledge management systems. We focused on these problems and identified six corresponding patterns: Publishing, Project-Document-Management, Skill Management, Support of Communities and Community building, Supporting Business-Processes-oriented knowledge support, Escalating Help-System. By giving an overview of these patterns, we can explicate how patterns can be related to each other and how a pattern language can evolve in a certain area such as knowledge management, and what kind of difficulties might occur. We focus on a selection of the description elements of a pattern (cf. section 1.2): title (1), introduction (3), problem-headline (4), problem-body (5), and solution (6). In all six cases, the corresponding larger pattern is the global knowledge management pattern which has been outlined above. Pictures, diagrams and references to smaller patterns are not included.

#### 1. Publishing

Publishing is used to distribute knowledge in a top-down process from the management to the employees to keep the latter informed about strategic considerations as well as rules and facts which are the basis for carrying out tasks.

*Problem:* Inconsistent, delayed, redundant distribution of knowledge.

It takes too much time for relevant new or changed information to reach the employees. Updating is awkward, information is inconsistent and is distributed across so many different locations that it is practically impossible to maintain consistency.

*Solution:* A team collects knowledge from management, departments and projects of the company and edits it. This includes updating, removal of redundancies and inconsistencies, focusing on essentials. An authoring tool is used to publish the information via intranet. This intranet must be the only source of information for employees. In order to avoid inconsistencies local copies of the sources should not exist.

#### 2. Project-Document-Management

The information circulation in and between projects is improved, an organizational memory is available for the project team.

*Problem:* Loss of information and time-consuming search or reconstruction of information.

The documents which are produced or needed by a project are not systematically stored from the beginning. Ideas and solutions have to be reinvented and are not available for related projects. Knowledge gets lost once people change teams.

*Solution:* The project management or team agrees on appropriate structures to store documents in a hierarchy of folders and to classify them according to certain categories. A central technical system is employed, such as Lotus Notes®, Livelink® or BSCW, to store the documents and conventions (versioning, distribution of access rights etc.) and established to keep the information up-to-date. Not only an editing team but every one has a responsibility.

#### 3. Skill Management

Companies want to know who knows what; therefore they develop a map of the distribution of knowledge within the organization.

*Problem:* No references to the tacit knowledge of experts.

Some knowledge (e.g. tacit knowledge, [26]) cannot be efficiently made explicit [24]. Large companies lose the overview over the knowledge of their personnel.

*Solution:* A database stores structured data on the employees' levels of experience, training, or expertise according to certain categories of competencies. Reoccurring queries are supported in a way that makes them easy to issue by any member within the organization. Skill profiles might be entered by the management, by the employees themselves or by a peer-to-peer process. Skill data bases do not need to contain explicit problem-oriented knowledge. However, user profiles can be built by registering the interest of people when they work with other systems, such as a project document-management component.

#### 4. Support of Community building

Web-technology supports the building of communities to develop and exchange knowledge or innovative ideas and learn together in a certain domain of interest or practice.

*Problem:* Limited occasions for collaborative building and exchange of knowledge.

Due to matters of time, separated locations or organizational structures, people are unable to meet with others to jointly elaborate their knowledge. Potentials for the building of social capital are therefore lost. *Solution:* Technical systems or components, such as newsgroups, up- and download support, chat etc., are used to enhance synchronous or asynchronous communication, question-answer-dialogues and the exchange of material. There is no editing team who maintains the developing content. The members of communities are informally, i.e. loosely coupled, and need not be formally bound (e.g. by a work contract). Relations emerge between actors with similar interests on a common topic. There are no strict rules (cp. the escalating help system) that guarantee the availability of certain supportive services.

#### 5. Process-oriented knowledge support

Storing and offering of knowledge is organized in accordance with business or work processes: to avoid an information over-

flow, information supply is adjusted to user, task and business-case characteristics.

*Problem:* Uncontrolled information overflow or information supply that does not comply with the task at hand.

The distribution of labor among knowledge providers and consumers frequently separates the maintenance of a knowledge base on the one hand from the task-related usage of knowledge on the other hand. Therefore, information is not considered in the cases where it could be helpful, and has to be searched, or is not stored at the appropriate position of the structured knowledge base.

*Solution:* Business and work processes are modeled. These models are represented in a technical knowledge base and partially enacted by a workflow component. The processes of knowledge handling are also modeled and combined with the processes of task completion [32], [38]. Knowledge can then be delivered and stored with regard to the task, which is processed at a specific moment, and with regard to the position of this task in the context of a process.

### 6. Escalating Help-System

A system provides knowledge-related help in problem situations on different levels. If the support on the first level is not sufficient, the next level can be activated and so on.

*Problem:* Knowledge or help are not offered when required

The processes of acquiring knowledge (e.g. in training sessions) and situations, where the knowledge is needed, are disintegrated. When problems occur, it is difficult or awkward to get access to people who could offer help.

*Solution:* Employees who have a problem should first search a content data base to learn whether they have a common problem which has already been solved. If not, they can publish their problem so as to receive help from other users or experts. Experts can be found with the help of a skill database; an editing team maintains the content data base.

Each of the six patterns combines proposals for technical components, for organizational structures and for conventions of human behavior. They represent socio-technical systems. The solutions cannot describe all the details required to bring them into reality, not even if there was more space available to extend the description in this paper. The meaning of the patterns becomes clearer only when they are put into a set of different patterns of knowledge management and by making the differences between them comprehensible.

### 3.2 Aspects of constructing a pattern: The example of the escalating help-system

The more details of a pattern are described, the more aspects and decisions have to be discussed which affect the quality of a pattern's description, and eventually the presented solution. To extend the description, Alexander [3] proposes the usage of a picture. In the case of knowledge management, this is not always a reasonable idea. However, it might be helpful to provide screenshots (such as fig. 4), a recorded sequence with a series of screenshots or a video clip – e.g. of a role play where the system is demonstrated. The other obligatory description element is a diagram – as shown in figure 3. In this subsection, we explain in detail what such a diagram might look like. It mainly serves two purposes: to discuss some of the details of the escalating help-system (without being complete) as well as the problems which are related to the detailed specification of a pattern.

Predecessors of the diagram have been used to support training of the users of the system, to explain to the management how the system could be improved, and to teach students who had to learn what knowledge-management could look like in a large company. The diagram is based on the modeling notation SeeMe as described at the beginning of this section. It shows that the usage starts with a task-related need for help. Fig. 3 indicates that certain features of a technical system are combined with the activities of people who have to take certain roles. There are indispen-

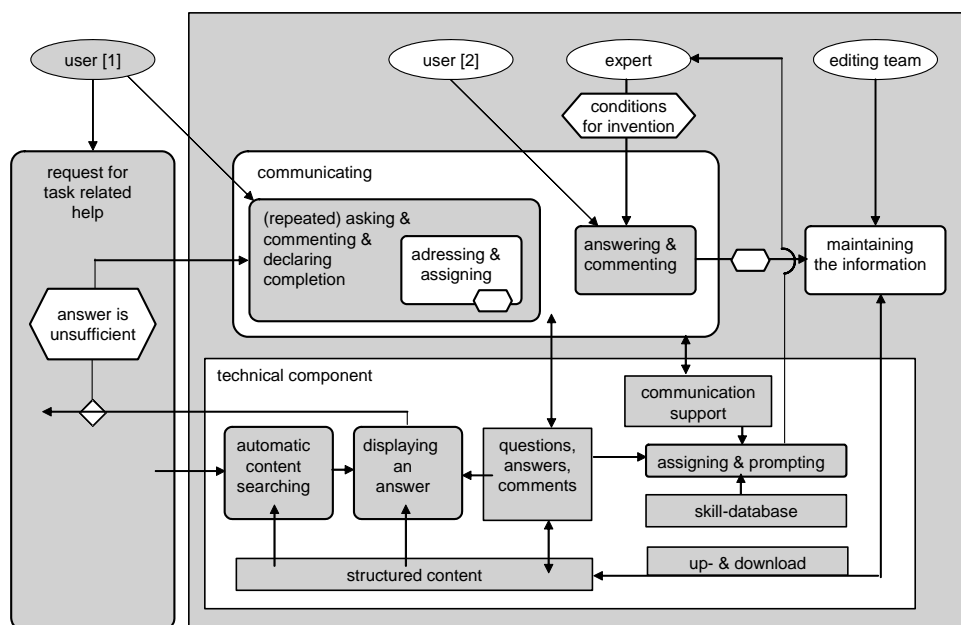


Figure 3: The escalating help-system

sable technical features and components required:

- stored, *structured content* which is searchable by the user or a search-machine to find answers to FAQs.
- *Communication support* which allows users to exchange questions, answers and comments and to insert them into the appropriate parts of the content. Furthermore, selective communications should be possible by choosing recipients who will then be able to read the contributions. The necessity of this feature was discovered during our empirical investigations. Subsequently, the need for an awareness mechanism became obvious that informs the addressees that a contribution has taken place.
- A *skill-database* should be employed – by using an appropriate matching algorithm – to assign requests to experts and to trigger them. The idea is inspired by the example of the AnswerGarden System [1] which can also be considered as a pattern.
- There should be an *up- and download support* so that the revision or the adding of new content can be made by the editing team.

The enumeration of these technical features makes it necessary to reflect questions which concern the general rational of describing a pattern. Representing technical features in a diagram is continuously accompanied by the question which details are excluded or included. Can an awareness mechanism – which is needed to inform other users about new questions – be considered as an integral component of *communication support* or do they have to be explicitly mentioned by using a graphical symbol. Is it necessary to display *up- and download* or is this an obvious feature of such a system? Wouldn't it be sensible to express that other features – e.g. an authoring tool – are not part of the problem solution? The problem with the degree of explicitness of a pattern is on the one hand that it should focus on essential elements and on the other hand that it should not be disguised whether elements, which might be expected, are included or not. This problem applies to text-based descriptions as well as to diagrams. However, it is advisable to combine both kinds of representation so as to achieve a basis for quality assurance.

The roles and activities of fig. 3 characterize the possibilities of escalating help. If the content search does not provide a sufficient answer, the question can be published in the system so that other users can offer their help. This employing of other users is left out in the first AnswerGarden concept [1]. However, our empirical investigation revealed that this option is effectively used and should therefore be integrated into the pattern<sup>1</sup>. If other users do not help, a group of experts have to intervene with respect to certain rules and conditions which have been determined by the company. For instance, the company has decided that every question which is published but not answered before 4 pm, has to be answered by an expert by the following morning. However, after about two months the experts started to gradually neglect this duty. Therefore, it seems to be reasonable to integrate a skill-database for selecting experts whose knowledge corresponds with

<sup>1</sup> AnswerGarden2 [2] follows a similar strategy: If a question is not answered, a component allows the user to consider whether to get answers from chat systems, bulletin boards, software agents, or other people.

a question. The selected experts can be automatically addressed and prompted to propose a solution.

It becomes clearer how a knowledge management pattern such as the escalating help system can be developed by combining experience from the literature [1] with concrete empirical results from certain cases. It is also obvious that such a pattern for a groupware application is highly dynamic and can never be considered as completed. It will certainly take several years to make groupware patterns as stable as architectural patterns.

Another characteristic of a socio-technical pattern are the conventions which have to be developed and complied with by the users. These conventions or socially constructed rules cannot be technically enforced but have to be 'implemented' by training, by making them explicit, and by the establishment of role models. Therefore, it is no surprise that these conventions were casually violated. Typical examples would be that a question should not be published before the stored content had been searched and that the users should declare a question as answered sufficiently so that others didn't have to offer further details.



Figure 4: Screenshot of a question answer dialogue

The diagram shows that the involved roles have a number of options to enable them to make free decisions, too. User[1] can decide whether he publishes a question or not and as to which audience it is made available. User[2] may choose as to whether he will intervene or not. A dialogue might eventually evolve. The editing team can then decide as to whether they analyze such a dialogue to improve the stored content or not. Due to these options there is not only one but a multiplicity of potential paths as to how the activities shown in the diagram can be combined. This multiplicity is a substantial reason as to why we adopt Alexander's approach of completing a pattern description by showing a diagram. The variety of possible instantiations of the pattern cannot be covered by using a text-based scenario description or by a single role play as was used to explain the escalating help system in the training session. The disadvantage of such diagrams is their higher level of abstraction. It is therefore advisable to combine them with textual descriptions and to add screenshots of the technical system. Fig. 4 gives an example of possible dialogue which might develop. The ticks indicate the questions which are already sufficiently answered. Thus, reminding the users, that they should not forget the step of *declaring completion*.

This example of a pattern shows that the technical components do imply a benefit but only if they are combined with appropriate user behavior. This emphasizes the necessity of describing both aspects in conjunction to each other, which is also necessary for the sub-patterns that need to be referred to (cf. no. 8 in Alexander's list).

**1. Title:** Preparation and involvement of the employees

**2. Pictures:** Presentation slides from the kick-off meeting, where all steps of the introduction process were explained

**3. Introduction with references to the larger patterns**

The strategy of introducing a knowledge management solution has influence on the acceptance of a new solution. Therefore a deliberate process as in the overall pattern “introduction of a socio-technical solution” is needed (in this paper see figure 1). In this pattern the preparation and involvement of the employees are described.

**4. Problem – headline**

Absence of qualification and motivation of the employees endangers the success of knowledge management solutions

**5. Problem – body**

A successful introduction of a knowledge management solution depends essentially on human-factors: on the motivation of the employees, their technical competence to use the new technology, or on their meta-knowledge that guides cooperative behaviors in using the system. These factors are often underestimated and not included in introduction strategies: Appropriate sharing or exchange processes, for instance, are not at hand, when the system is introduced. Usage conventions need time to commonly be developed. Acceptance does not grow unless a certain level of usage demonstrates the tool’s usefulness.

**6. Solution**

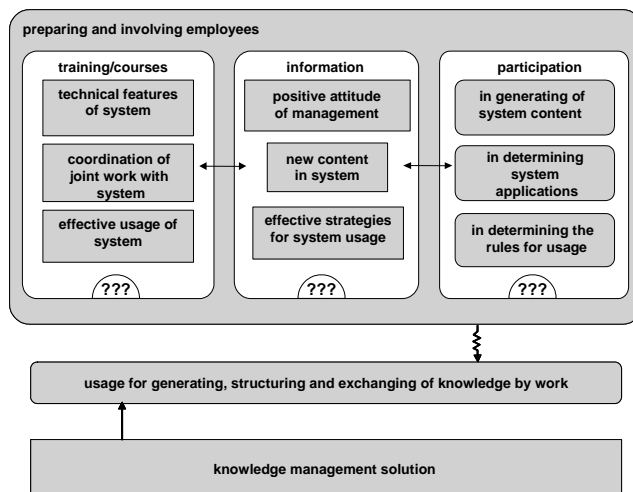
A company that wants to establish cooperative knowledge sharing on the basis of a knowledge management software must prepare employees by offering trainings in the beginning, by involving users in the contribution of contents and in the improvement of the technology, and, finally, by continuously providing feedback on the system’s use and usefulness.

**Initial Trainings** should not only cover the technical features of the solution but also strategies for the usage in every day, individual and joint work This is necessary, as a knowledge management solution should facilitate cooperation of many employees.

**Employee participation** is necessary as the solution should provide content that can only be articulated by the users themselves. Therefore participation in the generation of content and in the negotiation of rules and conventions for cooperation is required.

**Continuous information** is necessary for long term acceptance by the employees. It gives the feeling that the technical solution is a frequented and helpful tool. Topics for this informative activity are the positive attitude of the management, new content and either new or more effective procedures of collective system usage.

**7. Diagram**



**8. Final paragraph with references to smaller patterns**

This pattern can be accompanied by patterns for designing and conducting training courses or for the organization of participation processes.

**3.3 A pattern for the introduction of a solution**

Any knowledge management solution’s adoption and success depends on the way that it is introduced into the existing systems. We therefore see a need to develop patterns for the introduction of a technical solution in a company. Since they are used to shape the development of new processes in an organization, these patterns are referred to as organizational patterns [25]. While we look at both to create introduction patterns and to integrate social and technical aspects of the targeted solution in patterns of socio-technical systems, conventional organizational patterns exclusively focus on social structures and practices in “organizations that build (or use, or administer) computer software” [25]. The introduction patterns, found in our empirical research, represent best practices and can be used in various settings. In this subsection we look at details of the “Pattern for the preparation and involvement of the employees” as one example for our introduction patterns for knowledge management. The example uses Alexander’s format as described in section 1.

**3.4 Towards a pattern language for groupware applications**

A set of different patterns for groupware applications can be the basis for a pattern language. However, such a language needs a kind of internal structure which is based on the relationships be-

tween the patterns. Our examples make clear that there is a variety of different relationships which should be taken into account.

- A set of different patterns can be hierarchically structured. Typical relationships are those of aggregation or inheritance. For example every pattern discussed in 3.1 is in an “is-a-relation” to the general knowledge management pattern.
- A pattern or parts of a pattern are integrated into other patterns. Patterns can be overlapping – e.g. the skill management database in the case of the escalating help-system is also an element of the skill-management pattern. However, this pattern includes more elements, in particular the activities of keeping the skill descriptions up-to-date.
- Patterns can support or reinforce each other. For instance, one pattern can describe a procedure of how a solution described in another pattern can be put into practice (cf. the relation between pattern given above and the patterns in section 3.1 and 3.2) or can provide a means of reaching a solution described in another pattern (as the *Project-Document-Management-Pattern* does to projects aiming at the introduction of knowledge management systems). Due to the overlapping of patterns or the reciprocal support between them, it might be the case that certain patterns of a language are connected by a cyclic relationship. It is therefore not self-evident, that a pattern language will be structured as an acyclic, directed graph (by contrast, cp. [6]).
- Patterns supplement each other to achieve a broader solution.

- e) There are building blocks which are part of more than one pattern which might have their own position or a specific function in a pattern language, although they do not represent a complete solution of a real problem. Typical examples are patterns that describe the structuring of contents in a hierarchical fashion or patterns for the composition of data displays.
- f) Patterns can be ordered by using various criteria, for instance how they could occur in the course of a larger process or with respect to other coordination mechanisms such as being related to certain resources, time slots etc. [21]
- g) Patterns could be grouped using their characteristics as a basis – e.g. depending on the question as to whether their design was driven by technical progress or organizational and social requirements.
- h) Patterns can hold contextual relationships to other patterns that only become apparent, if they are compared directly with each other. If patterns are similar but have different salient aspects, it is useful to bring them into a direct relationship so as to make them easier to understand. This is also relevant for the six knowledge management patterns described above. The particularity of each of them would not be properly understood if they were presented in isolation.

The variety of possible relationships between individual patterns shows us that a pattern language needs a carefully designed architecture within itself. In addition to the existing web-based systems for publicly collecting patterns and supporting shepherding procedures, procedures for creating pattern languages are also needed. Many years of practical experience within the field for which a pattern language is to be constructed is essential to fulfill this task.

#### 4. DISCUSSION

We have traced the history of patterns and pattern languages from their origins in urban architecture to their application in different domains of computer science. A first account of important characteristics of patterns in the domain of groupware applications was given and enhanced by examples taken from our experience in the field of knowledge management systems. Based on six patterns within this domain, our argument is that the step from interrelated patterns to a coherent pattern language does not just occur but requires much effort.

Since knowledge management includes a lot of typical aspects of groupware, and since our theoretical and practical work has benefited from the patterns described above, our conclusion is that the concept of patterns is indeed valuable for CSCW and groupware applications. The above view on patterns for knowledge management is promising for science and practice with groupware applications. The advantage of patterns is that they have an empirical basis and therefore represent ground knowledge. This can support companies in two different ways: one is to show possibilities for the usage; the other is a recommendation for a specific process.

The patterns for socio-technical solutions as described in 3.1 offer different alternatives. They give an overview of potential support which helps to select the solution which then might be helpful for the specific problem of a company. Being socio-technical patterns they not only show technical aspects (e.g. what are essential functionalities of the technical solution?) but also social factors (e.g. who interacts with whom? How does the technical system sup-

ports the interactions?). Therefore they can guide organizational design, too.

The introduction patterns of figure 1 can be understood as a recommendation for the introduction of a knowledge management solution. It is of great relevance for companies which introduce a new technical system and want to prepare the employees. By following the patterns, companies can benefit from the experience made by other organizations and improve the acceptance of the employees.

Nevertheless there are potential risks in blindly applying the idea of patterns. Some are related to well-known discussion of formalizing and categorizing working procedures. Abstracting from a single solution to an even more abstract concept, which is generic to a class of solutions is even more problematic, since there is less of the rich organizational reality reflected in the representation.

Other problems are connected to the concept of patterns themselves. In the field of urban architecture, Alexander and his colleagues were able to draw from cultural experiences that were more than a thousand years old. Manifold examples can be found, and underlying principles (the “quality without a name” in terms of Alexander) can be derived from a broad range of settings. Explicating general principles and patterns in a young field such as CSCW is more problematic. Solutions have not yet often been employed. They have only been in use for a relatively short period of time. These limitations would suggest that it is better to use patterns as reflective artifacts rather than as “installation guides” for socio-technical systems.

Considering all this information, we think that the domain of groupware applications is mature enough to create patterns and pattern languages that have an instructive message as in the Alexanderian meaning. Based on what we have shown in respect to knowledge management, we could say

- that manifold types of relationships exist between patterns.
- that introductory patterns need to be combined with patterns describing socio-technical solutions.
- that the meaning of a pattern can only be grasped in context to patterns in its direct neighbourhood.
- that social and technical aspects must be integrated within each pattern.

Authors of patterns must be sure to link them to empirical data; it is not sufficient to have a design, test it on one single case and then publish a pattern. Adhering to Alexander we must not forget that “The pattern is, in short, at the same time a thing, which happens in the world, and the rule which tells us how to create that thing, and when we must create it.”

#### REFERENCES

- [1] Ackerman, M. S. Augmenting Organizational Memory: A Field Study of Answer Garden. *ACM Transactions on Information Systems* 16 (3), 1998, 203–224.
- [2] Ackerman, M. S. Answer Garden 2: Merging Organizational Memory with Collaborative Help. *Proceedings of CSCW'96*, 97-105.
- [3] Alexander, Ch.; Ishikawa, S; Silverstein, M.; Jacobson, M; Fiksdahl-King, I.; Angel, S. *A Pattern Language. Towns*

- Buildings Construction*. New York, Oxford University Press, 1977.
- [4] Alexander, Ch. *The Timeless Way of Building*. New York, Oxford Univ. Press, 1979.
- [5] Bannon, L. J. The Politics of Design: Representing Work. *Communications of the ACM* 38(9), 1995, 66-68.
- [6] Borchers, J. O. A Pattern Approach to Interaction Design. *AI & Society* 15, 2001, 359-376.
- [7] Bowers, J. The politics of formalism. In: Lea, M. *Contexts of Computer-Mediated Communication*. New York, Harvester Wheatsheaf, 1992, 232-261.
- [8] Dearden, A.; Finlay, J.; Allgar, E.; McManus, B. Using Pattern Languages in Participatory Design, *Proceedings PDC 2002* (Malmö, SWE, 23-25 June 2002), 104-113.
- [9] Diefenbruch, M.; Hoffmann, M.; Misch, A.; Schneider, H. Situated Knowledge Management - on the borderline between chaos and rigidity. *Proceedings of PAKM 2000* (Basel, CH, October 2000), 8-1-8-7.
- [10] Diefenbruch, M., Herrmann, Th.; Hoffmann, M.; Schaal, H.-G. (eds.): Workshop Erfolgsfaktoren bei der Einführung von Wissensmanagement-Lösungen. *Proceedings GI-Jahrestagung 2002* (Dortmund, GER, 1-3 Oktober, 2002), Bonn: Lecture Notes in Informatics, 695-736.
- [11] Eason, K. *Information Technology and Organisational Change*. Taylor and Francis, 1988.
- [12] Erikson, T. Lingua Francas for Design: Sacred Places and Pattern Languages. *Proceedings of DIS 2000* (Brooklyn, NY, August 17-19, 2000). New York, ACM Press, 357-368.
- [13] Gamma E.; Helm R.; Johnson R.; Vlissides J. *Design Patterns: Elements of Reusable Object-Oriented Software*. Reading, Mass, Addison-Wesley. 1995.
- [14] Groupware Patterns. <http://swiki.darmstadt.gmd.de/gw-patterns/>
- [15] The HCI Patterns Pages. <http://hci.ethz.ch/patterns/>
- [16] Herrmann, Th.; Loser, K.-U. Vagueness in models of socio-technical systems. *Behavior & Information Technology* 18(5), 1999, 313-323.
- [17] Herrmann, Th.; Hoffmann, M.; Kunau, G.; Loser, K.-U.: Modelling Cooperative Work: Chances and Risks of Structuring. In: Blay-Fornarino, M. et al.: Cooperative Systems Design, A Challenge of the Mobility Age (*Proceedings of Coop 2002*). Amsterdam: IOS Press. S. 53-70.
- [18] Herrmann, Th.; Mambrey, P.; Shire, K. (eds.): *Wissensgenese, Wissensenteilung und Wissensorganisation in der Arbeitspraxis*. Wiesbaden: Westdeutscher Verlag. 2002.
- [19] Herrmann, Th.; Kienle, A.; Reiband, N.: Meta-knowledge - a success factor for computer-supported organizational learning in companies. *Educational Technology & Society*, 6(1), 2003, 9-13.
- [20] Lancaster's PoInter Project: <http://www.comp-lancs.ac.uk/computing/research/cseg/projects/pointer/pointer.html>
- [21] Malone, Th. W.; Crowston, K. What is Coordination Theory and How Can it help design cooperative work Systems? *Proceedings of CSCW'90* (Los Angeles, CA, 7-10 October, 1990), 357-370.
- [22] Martin, D.; Rouncefield, M.; Rodden, T.; Sommerville, I.; Viller, S. Finding patterns in the fieldwork, *Proceedings of ECSCW 2001* (Bonn, GER, 16-20 September, 2001), 39-58.
- [23] Mumford, E. A Socio-Technical Approach to Systems Design. *Requirements Engineering* 5(2), 2000, 125-133.
- [24] Nonaka, I.; Takeuchi, H. *The Knowledge Creating Company*. New York, Oxford, Oxford University Press, 1995.
- [25] Organizational Patterns. <http://www.easycorp.org/cgi-bin/OrgPatterns?OrganizationalPatterns>.
- [26] Polanyi, M. *Tacit Dimension*. London, Routledge & Kegan Paul Ltd. , 1966.
- [27] Probst, G. *Practical Knowledge Management*, Prism, Arthur D. Little, Second Quarter, 1998.
- [28] Robinson, M.; Bannon, L. Questioning Representations. *Proceedings of ECSCW'91* (Amsterdam, NL, 24-27 September, 1991), 219-233.
- [29] Scheer, A.-W. *Architecture of Integrated Information Systems, Foundations of Enterprise Modeling*. Berlin, Springer, 1992.
- [30] Schmidt, K. Of maps and scripts - the status of formal constructs in cooperative work. *Information and software technology* 41, 1991, 319-329.
- [31] Schuler, D. A Pattern Language for Living Community. *Proceedings PDC 2002* (Malmö, SWE, 23-25 June 2002), 434-436.
- [32] Staab, S.; Studer, R.; Schnurr, H.-P.; Sure, Y. Knowledge processes and ontologies. *IEEE Intelligent Systems* 16(1), 2001, 26-34.
- [33] Suchman, Lucy A. *Plans and situated actions: The problem of human-machine communication*. Cambridge U.K., Cambridge University Press, 1987.
- [34] Suchman, L. Do Categories Have Politics? The language/action perspective reconsidered. *Computer Supported Cooperative Work* 1994(2), 177-190.
- [35] Suchman, L. Making Work Visible. *Communications of the ACM* 38(9), 1995, 56-64.
- [36] Thomas, J.C.; Danis, C. M.; Lee A. *Who Speaks for Wolf?* Report of the T.J. Watson Research Center. RC22644 (W0211-186) November 19, 2002.
- [37] UI Patterns and Techniques. <http://timetripper.com/uipatterns/>
- [38] Wargitsch, C.; Wewers, Th.; Theisinger, F. An Organizational-Memory-Based Approach for an Evolutionary Workflow Management System - Concepts and Implementation. *Proceedings HICSS'31*, Vol. I, (Los Alamitos, 1998), 174-183.

Lotus Notes is a registered Trademark of IBM.

Livelihood is a registered Trademark of Open Text Corporation.