

# Constructing a Groupware Pattern Language

## Positionpaper for EuroPLoP 2002 Focus Group: What makes Pattern Languages work well?

**Torsten Holmer**

Fraunhofer Institute Integrated Publication and Information Systems  
Dolivostr. 15, 64293 Darmstadt Germany  
holmer@ipsi.fhg.de

### **The Idea of Groupware Patterns**

The research area of CSCW (computer-supported cooperative work) and the development of groupware (software to support groups) deals with people, tasks and technologies and how to put them together in a way that helps people to reach their goals better than without computer support. It is a very young field of computer science and many of its concepts are still in a kind of a prototype state. Although there is much need for further research, the community has gained some experience in at least partially successful groupware design. But many good ideas stayed research prototypes and the commercial market is often far beyond the technical possibilities that were discovered by CSCW research. We see one reason for that in the case that the proven knowledge is widespread and unsorted. As known from other areas, patterns have proven to solve these problems to some degree.

We are starting a project on groupware patterns, in which we try to collect patterns for groupware design and development. The long-term objective is to create a collection of patterns, which should guide developers and customers in the process of creating groupware.

Because the area of CSCW and groupware is very complex, the gathering of the patterns should be an activity of the CSCW community. As a starting point we defined some patterns which we will refine step by step. We also started to create a pattern map to create relationships between the proposed patterns.

### **The need for a Pattern Language**

As we started to create the patterns map, we were confronted with the task to organize a very complex structure. We discovered that the simple collection of patterns leads to an unsorted list. So, we have the words but we could not speak sentences. We tried to identify families (groups of related patterns) and were to some degree successful, but we still feel that the structure is not finished or at least comprehensive. Especially the function of guiding the design of groupware needs a very structured collection where the designer and other users are lead on a meaningful path through the collection. Our hope is that a Pattern Language would be the right

structure for our goal to guide experts and novices in the process of developing groupware. We assume that we even need more than one language because the design process covers different aspects:

- discussions with customers/users (process and organizational patterns),
- designing the application (groupware design patterns)
- implementing the application (groupware implementation patterns)

Beneath structuring the languages themselves there is also the task of linking them together, so that the group of developers running a groupware project, can go from the requirement analysis to the design and then to implementation phase. The core of this activities are still the groupware design patterns because they are hosting the central concepts.

### **How to create a Pattern Language?**

From the work of Christopher Alexander we know that he had organized his patterns on different levels which point to different architectural design areas. Some patterns are related to towns, some to city areas and some to the design of houses and rooms. The level of granularity and therefore some kind of hierarchy was the guiding principle in this language. Patterns were part of other patterns or contain other patterns. The design of groupware applications could be organized in the same way. There are some choices and decisions on a very general level, which lead to more detailed options and design situations. General decisions are about synchronous and asynchronous use, the size of the cooperating groups and the intended network connection or security and privacy constraints. More detailed design questions are the design of single windows and which elements should be put where.

Our approach for creating a pattern language for groupware design patterns will be to look for these relationships and how we could reorganize our patterns map.