Towards an ontology of collaboration patterns

Jonas Pattberg, Matthias Fluegge

Fraunhofer Institute for Open Communication Systems Fokus
Kaiserin-Augusta-Allee 31
10589 Berlin
jonas.pattberg@fokus.fraunhofer.de
matthias.fluegge@fokus.fraunhofer.de

Abstract: The concept of patterns and pattern languages has been applied in different application domains like software engineering, human computer interaction, and pedagogy. In the area of Collaborative Working Environments (CWE) there are different understandings on what collaboration patterns are and how they can be described and defined. Collaboration patterns are specified at different levels of granularity and in relation to different application contexts. In this article, after introducing the general idea of patterns and its application in the CWE domain, we present an approach for creating a layered ontology in order to integrate collaboration patterns of different granularities and at different levels of abstraction.

1 Introduction

The concept of patterns and pattern languages has been adopted in the domain of Collaborative Working Environments (CWE) resulting in so called collaboration patterns. Collaboration patterns can be regarded as a kind of documentation [He03] for a proven collaboration solution. Furthermore, collaboration patterns act as a 'lingua franca' [Er00], as a common language for all involved stakeholders: designers, developers and users of CWEs. As part of the European project EACE[1] – "Expediting the Adoption of Collaborative e-working Environments" a thorough review of the relevant literature has been carried out, finding quite different interpretations of collaboration patterns. In this paper, after introducing the idea of patterns and its application in the CWE area, we present an approach for creating an ontology of collaboration patterns and patterns languages, thus consolidating different views and concepts in this area.

[1] This work is supported by the European Commission through the Specific Support Action "EACE" (FP6 – 022585).

2 Patterns and pattern languages

"*Each pattern describes a problem that 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.*" [Al77]

This citation of the architect Christopher Alexander, who started using patterns and pattern languages to describe the events and forms that can be found in cities, towns and buildings, was introduced to architecture in the 1970's. For example, Alexander in [Al88] uses his pattern language to define a new planning process for the University of Oregon. He states, that each pattern is a three-part rule, which expresses a relation between a certain context, a problem and a solution.

2.1 Patterns in computer science

In software development an analogous situation can be found. Various software systems frequently use common concepts to solve recurring problems. In 1987, Beck and Cunningham began applying the idea of patterns to programming and presented their results at the OOPSLA (Object-Oriented Programming, Systems, Languages and Applications) conference [BC87]. Gamma et al. published a book of design patterns for object-oriented software engineering [Ga95] and advanced the popularity of patterns in computer science. In the same year the annual "Pattern Languages of Programming" (PLoP) conference was established, providing a forum for exchanging patterns of recurring software design problems. The goal of applying patterns in software engineering is to simplify the process of building increasingly complex systems by providing tested, proven development paradigms. Reusing design patterns helps to prevent subtle issues that can cause major problems and it improves code readability for software engineers and architects who are familiar with these patterns. Apart from software development, the idea of patterns has been adopted in the HCI (human computer interaction) area. In [Bo00] Borchers describes patterns for designing user interfaces in a heterogeneous team of interface designers, including application domain experts and developers. In [Ti05] Tidewell describes various design patterns for interfaces. The application area of these interfaces ranges from desktop applications, web sites, and web applications to mobile devices. An online repository of design patterns can be found at the Yahoo! User Interface Library [Ya07].

2.2 Patterns in education

"*The intent [of pedagogical patterns] is to capture the essence of the practice in a compact form that can be easily communicated to those who need the knowledge. Presenting this information in a coherent and accessible form can mean the difference between every new instructor needing to relearn what is known by senior faculty and easy transference of knowledge of teaching within the community*" [Be07]

Fricke and Völter [FV00] developed a pattern language for teaching seminars. They visualise the patterns using a "pattern map". Fricke and Völter identified 48 patterns from "Adapt To Participants Background" over "Nameplate" to "Separate Similar Content". These patterns are intended to help instructors, who are not studied educators, to improve their seminars by providing proven techniques for running a good seminar.

The pedagogical pattern project [Be07] has a similar objective. It aims at helping teachers to teach and students to learn, since many trainers and educators were not taught to teach, but are rather teaching by 'accident'. The project collects pedagogical patterns as a method for capturing and communicating knowledge about the principles and methods of teaching. The "Person-Centred e-Learning" (PCeL) Pattern Repository [De07] is a blended learning approach based on humanistic educational principles. It describes common PCeL practices in a uniform, structured, and partly visual format. The project is based on a PhD research [De05] and provides patterns in different 'packages' ranging from 'Course Types' over 'Interactive Elements' to 'Project Based Learning'.

2.3 Concepts of pattern languages

Christopher Alexander also shaped the term "pattern language" [Al77]. A pattern language is a set of interrelated patterns. The interrelations represent dependencies and refinement relations. Other classifications of patterns are categories, 'families' [Ga95, Ti05, Ya07, Sc07, Sch07] and lists of patterns [Ma07] as informally related patterns enumerations. Additionally, some Authors visualise the pattern language by means of a pattern map [FV00, Sc07].

3 Concepts of collaboration patterns

In the following we will provide an overview of different research works and projects dealing with the concept of patterns and pattern languages in the CWE area.

Molina, Redondo, and Ortega use pattern-based techniques to design groupware systems. They propose a design and development process based on the use of several conceptual models. In this process several techniques are used. In [MRO06] collaboration patterns for modelling collaborative tasks and protocols of cooperation are defined. It is stated that, due to their multi-disciplinary nature, the development of Computer Supported Collaborative Work (CSCW) systems is not a trivial task. Additional problems arise from the social and distributed nature of such systems. Molina, Redondo, and Ortega introduce a methodology, named CIAM (Collaborative Interactive Applications Methodology) adopting different viewpoints for creating conceptual models of CSCW systems.

The Liberating Voices! Pattern Language Project [Sc07] aims to "help understand, motivate and inform the worldwide movement to establish full access to information and communication - including the design and management of systems to build information and communication technology that is democratic, inclusive and meets human needs". The project invites interested people and connects practitioners and researchers, community organisers and policy-makers. The project uses the idea of patterns and a pattern language to provide a useful "knowledge structure" that represents the collective wisdom and aspirations of the community. A central idea is that, while individual patterns might be useful, the structure of a pattern language will make it easier to integrate the patterns into a coherent whole. The goal of the project is to develop one or more pattern languages which can help people to think about, design, develop, manage and use information and communication systems. The Liberating Voices project defines "pattern language" as a holistic collection of "patterns" intended to be used together to address a problem. The project selected more than 240 patterns which are published on the project website. The patterns are organised in different themes and categories, including collaboration. The Patterns4Groupware project [Sch07] maintains a comprehensive online catalogue of groupware patterns. The pattern language consists of families of related patterns. Each pattern provides proven solutions for a groupware problem, and it is expressed independently from the underlying technology. For authors interested in submitting patterns, a "Swiki" authoring tool providing a corresponding pattern template is provided. Each pattern is described by means of the following properties: Name, intent, family, problem, scenario, context, indications, solution, participants, rationale, safety rules, known uses, related patterns, status, authors, e-mail, and references. The project also assists authors, interested in contributing patterns, by means of workshops and 'shepherds'. The workshops provide a chance to discuss new patterns, and the 'shepherds' use their experience to review the patterns. The workshops and computer mediated discussion groups try to involve interested people and to establish a community. All patterns are presented in a uniform way by means of a web-based catalogue [Sch02]. The Pointer (Patterns Of Interaction) project at the Lancaster University [Ma07] identifies "regularities in the organisation of work, activity, and interaction amongst participants, and with, through, and around artefacts" [Ma03]. The project presents descriptive patterns of ethno-methodically informed ethnographic studies of work and technology. The project work is based on field studies to describe how people are acting and interacting in 'configuration' with computers and other artefacts to achieve their work. Each identified pattern is linked to individual examples or vignettes from particular pieces of these field studies. The declared rationales behind this work are:

- Re-use of knowledge
- Description and comparison of similar work arrangements
- Resources for multiple stakeholders
- Quality check whether artefacts really work and fit with social circumstances

The project considers patterns as something between design and examples. In contrast to Alexander, patterns neither are seen as a kind of rule nor as a solution for the 'right' configuration and practice. However, the project considers the patterns as a basis of studies with associated findings that may help designers of artefacts which support collaboration [Ma03].
Walton introduces in [Wa99] different types of dialogues (cp. Table 3-1). These are part of an argumentation theory "the new dialectic", a framework for evaluating arguments. It is opposed to "positivistic" philosophy / argumentation with only deduction and induction as means of reasoning. Walton differentiates between six types of dialogues (persuasion, inquiry, negotiation, information seeking, deliberation and eristic), different contexts, individual goals of the participant, and different goals of the dialogue. The argumentation theory is based on presumptive reasoning with a tentative basis as subject to commitments or retraction. It provides different proof standards (25 proof standards in [Wa96]) with critical questions and burden-of proof. The framework is applicable for everyday argumentation, evaluating argumentation, medical, legal, and ethical argumentation, computer science and general collaborative dialogues.

Type of Dialogue	**Initial Situation**	**Participant's Goal**	**Goal of Dialogue**
Persuasion	Conflict of opinions	Persuade other party	Resolve or clarify issue
Inquiry	Need to have proof	Find and verify evidence	Prove (disprove) hypothesis
Negotiation	Conflict of interests	Get what you most want	Reasonable settlement
Information seeking	One party lacks information	Acquire or give information	Exchange information
Deliberation	Dilemma or practical choice	Co-ordinate goals or actions	Decide best course of action
Eristic	Personal Conflict	Verbally Hit Out at Opponent	Reveal Deeper Basis of Conflict

Table 3-1, Classification of Dialogue [Wa99]

4 Ontology of collaboration patterns

In the EACE project a multitude of collaboration patterns and collaboration pattern languages was collected from literature, research projects and from domain-specific scenarios. Some survey results were presented in the previous sections. From the analysis it becomes clear that there are different understandings on what collaboration patterns are and how they can be described and defined. It turned out that people define collaboration patterns at different levels of granularity and in relation to different application contexts. What some people call "collaboration patterns" others call "collaboration services". Moreover, the more detailed one observes the collaboration of people, the more patterns (virtually an unlimited number) can be identified. In order to gain a coherent view on this topic it is therefore necessary to consolidate and categorise the patterns and to point out their interrelations. Therefore, an approach for building an extensible pattern ontology is needed.

In the following, rather than trying to design an "all-embracing" ontology, a generic concept for integrating collaboration patterns of different granularities is provided.

The proposed structure for an ontology corresponds to a "collaboration stack" (cp. Figure 4-1) that identifies the various levels of abstraction, ranging from abstract collaboration patterns on the one side to collaborative services and communication technologies on the other side.

The "collaboration stack" is inspired by a typical IT protocol stack. It clarifies the relation of collaboration patterns to collaborative services and to the underlying communication technologies. The collaboration patterns make use of the underlying collaboration services, such as shared workspaces, whiteboards, forums, etc. These collaboration services in turn make use of basic communication technologies, such as (wireless) networks, WWW, telephone networks, etc. On the other hand, low level (i.e. simple) collaboration patterns are used to build high level (i.e. advanced) or even abstract collaboration patterns. One may discuss controversially, whether in the Alexandrian sense, the notion of an 'abstract' collaboration pattern is admissible, since patterns should always contain guidance on how to solve a problem in a context. We think, however, that a collaboration pattern may be described in such a generic or coarse-grained manner that it needs refinement to be of real value, i.e. to be instantiated. Abstract collaboration patterns act as classifications and therefore add value to the ontology. All patterns together build a (layered) pattern language.

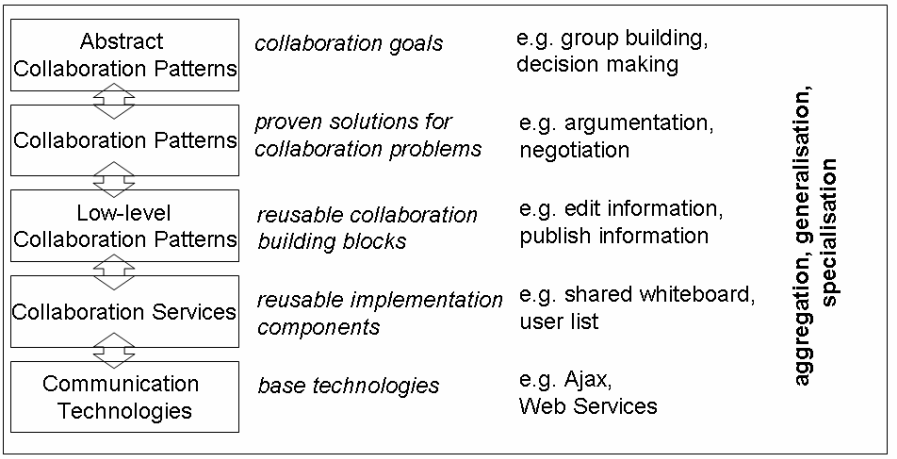

Figure 4-1: Collaboration Stack

Principles of the collaboration stack are:

- Each layer provides "interfaces" to the super- or subjacent layer.
- The layers "communicate" only through the provided interfaces. Therefore, only adjacent layers are directly related.
- Collaboration patterns are not collaboration services, but they make use of them. Therefore they may influence the design of new collaboration services, and new collaboration services may influence existing collaboration patterns, or enable the creation of new ones.
- The same applies to communication technologies: The emergence of new communication technologies may put forth new collaboration services. On the other hand, the need for certain collaboration services may influence the development of adequate communication technologies.
- Patterns can be composed to advanced patterns (aggregation).
- Patterns can inherit from each other.

In literature several attempts for providing a classification of abstract (high-level) collaboration patterns can be found. Therefore, when designing the ontology part that corresponds to the top level of the collaboration stack, such existent classifications should be considered and consolidated. Figure 4-2 illustrates how this can be done. Here de Moor's classification of collaboration patterns is combined with Andriessen's classification of group tasks. In [Mo06] de Moor classifies patterns according to the following categories:

- Goal patterns represent objectives of a community and of individuals.
- Communication patterns describe communicative interactions within a community.
- Information patterns conceptualise knowledge obtained from knowledge analysis activities. They define the knowledge elements essential for the collaborative process.

- Task patterns define which information patterns are associated with particular steps in the communicative process, thus describing the role, content can play in collaborative communication.
- Meta-patterns are conceptual patterns necessary for interpreting, validating, linking, and assessing the quality of other collaboration patterns.

In [An03] Andriessen develops a classification of the tasks a group can perform. For the classification two dimensions of characteristics are considered: The x-axis indicates whether the task is more cognitive or behavioural in nature. The y-axis indicates the degree of collaboration required for the task. Together with the four quadrants (Generate, Choose, Execute and Negotiate) the following task categories are identified:

- Creativity tasks, e.g. generating ideas
- Planning tasks, e.g. generating plans
- Performance tasks, e.g. executing performance tasks
- Contests / competitive tasks, e.g. resolution power conflicts
- Mixed motive tasks, e.g. resolving conflicts of interests
- Cognitive conflict tasks, e.g. resolving conflicts of viewpoints
- Decision making tasks, e.g. deciding without right answers
- Intellective tasks, e.g. solving problems with right answers

The extension of de Moor's classification of collaboration patterns with Andriessen's classification of tasks is illustrated in Figure 4-2. As already mentioned, this classification neither is regarded as an exclusive nor as an exhaustive classification of abstract collaboration patterns. Rather, it demonstrates the consolidation of existing high-level pattern classifications and it may be extended to meet more specific demands.

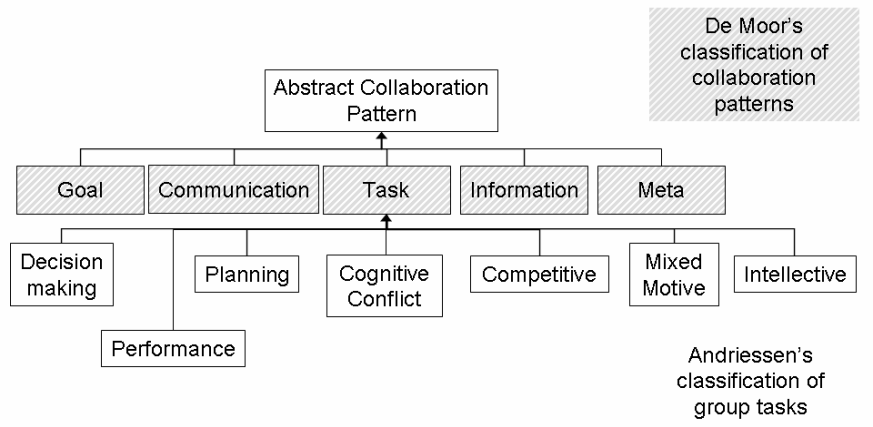

Figure 4-2: Classification of Collaboration Patterns

Figure 4-3 demonstrates how an ontology of collaboration patterns can be built that corresponds to the structure of the collaboration stack shown in Figure 4-1. In the provided example, the upper four layers of the stack are reflected[2]. Each layer directly makes use of the patterns from the adjacent lower level. The top level covers the 'abstract collaboration patterns' and their relations according to the above discussed consolidation of existing classifications. As these patterns are highly generic, they build on the 'collaboration patterns' identified in the second layer. 'Low level collaboration patterns' are fundamental building blocks that are used by most of the superior patterns. The 'collaboration services' identified at the ontology level below cover technical aspects and facilities that are needed to technically implement collaboration patterns.

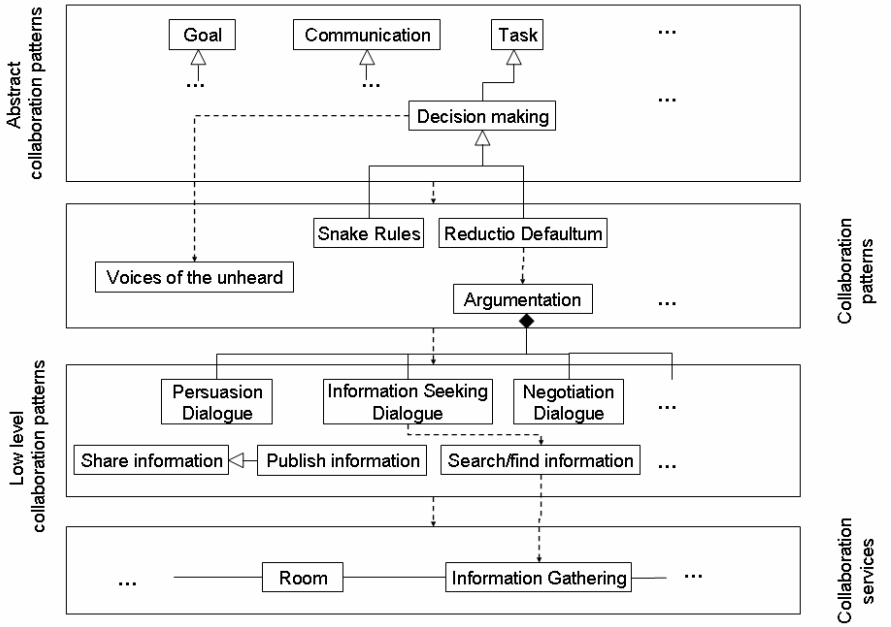

Figure 4-3: Ontology of collaboration patterns (excerpt)

An example clarifies the relations between the different levels. For instance, the 'abstract' collaboration pattern 'ReductioDefaultum' is a refinement of the 'Decision making' pattern. In order to implement it, the 'Argumentation' pattern, i.e. a composition of diverse dialogue types, is used. The dialogue type 'Information seeking' in turn uses the 'low level' collaboration pattern 'Publish information'. The 'Publish information' pattern builds on the collaborative service 'shared workspace'.

[2] The patterns from this example originate from various sources: "SnakeRules" and "ReductioDefaultum" are described in [Ch05, Ch05a], "Voices of the Unheard" were taken from [Th01], "Persuasion Dialogue", "Information Seeking Dialogue" and "Negotiation Dialogue" are mentioned in [Wa99], "Room" is taken from [Sch03] and revised in [SL07], and "Information Gathering" originates from [De05].

The proposed ontology structure is extensible and thus suitable to cover a broad variety of patterns. Through its layered architecture it enables the consolidation of different views (regarding granularity and purpose) on collaboration patterns.

5 Conclusions and future work

The development of collaboration patterns still is in an early stage. There is a growing community applying the concept of pattern and pattern languages in human collaboration and 'social-technical' systems. Several methodologies for identifying patterns and different understandings of collaboration patterns result in various pattern languages. Collaboration patterns are published in diverse online catalogues and they are subject to a number of conferences. Moreover, collaboration patterns partially cover different application domains and they vary in the covered extent of the collaboration.

The presented approach aims at consolidating different concepts of collaboration patterns and pattern languages by means of a layered ontology which is not limited to a small collaboration subset or to a single application domain. Rather, it is open for any kind of collaboration pattern, integrating them with different levels of abstraction.

Such an ontology extends the concept of pattern languages as it provides different levels of abstraction. Thus, it could serve as a kind of lingua franca which helps to improve communication between different stakeholders, ranging from users to CWE designers and developers.

Having defined the general structure of the ontology, future work will include its refinement and extension according to the patterns collected from CWE-related literature and projects. Potential refinements of the ontology may add attributes to the entities, such as origin or version/maturity of a pattern.

References

[Al77] Christopher Alexander et al. A Pattern Language: Towns - Buildings - Construction. Oxford University Press 1977.

[Al88] C. Alexander, M. Silverstein, S. Angel, S. Ishikawa, and D. Abrams. The Oregon Experiment. Oxford University Press, 1988.

[An03] J. H. Andriessen. Working with groupware: understanding and evaluating collaboration technology. London. Springer Verlag. 2003.

[BC87] K. Beck, Ward Cunningham. Using Pattern Languages for Object-Oriented Program. Submitted to the OOPSLA '87 workshop on the Specification and Design for Object-Oriented Programming.

[Be07] Joseph Bergin et al., The Pedagogical Patterns Project: http://www.pedagogicalpatterns.org/

[Bo00] Jan O. Borchers. A pattern approach to interaction design. Proceedings of the conference on Designing interactive systems: processes, practices, methods, and techniques. ISBN 1-58113-219-0, New York, P. 369-378, ACM Press, 2000.

[Ch05] Jonathan Cheyer (BlueOxen). "ReductioDefaultum". Available online at: http://collab.blueoxen.net/cgi-bin/wiki.pl?ReductioDefaultum. 2005.

[Ch05a] Jonathan Cheyer (BlueOxen). “SnakeRules”. Available online at: http://collab.blueoxen.net/cgi-bin/wiki.pl?SnakeRules. 2005.

[De05] Michael Derntl. Patterns for Person-Centered e-Learning. PhD Thesis. University of Vienna. 2005. URL: http://elearn.pri.univie.ac.at/derntl/diss/diss-derntl.pdf

[De07] Michael Derntl et al., The Person-Centered e-Learning Pattern Repository: http://elearn.pri.univie.ac.at/patterns/

[Er00] T. Erikson. Lingua Francas for Design: Sacred Places and Pattern Languages. Proceedings of DIS 2000 (Brooklyn, NY, August 17-19, 2000). New York, ACM Press, pp. 357-368.

[FV00] Astrid Fricke and Markus Völter, SEMINARS A Pedagogical Pattern Language about teaching seminars effectively. EuroPLoP 2000 conference.

[Ga95] Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides. Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley. ISBN 0-201-63361-2. 1995.

[He03] Thomas Herrmann, et al. Concepts for Usable Patterns of Groupware Applications. GROUP’03, November 9–12, 2003, p 349 ff.

[JH04] Erik Johnston, Darrin Hicks. Speaking in teams: motivating a pattern language for collaboration. Interdisciplinary Description of complex systems, 2 (2), p.136-143, ISSN 1334-4676, 2004.

[Ma03] David Martin. Patterns Of Cooperative Interaction – A Brief Introduction To The Lancaster Perspective. From Good Practices to Patterns. Mining socio-technical patterns from experience with groupware. A workshop at ECSCW 2003 in Helsinki, Finland, 2003.

[Ma07] David Martin, et al., The PoInter Project: http://www.comp.lancs.ac.uk/computing/research/cseg/projects/pointer/patterns.html

[Mo06] Aldo de Moor. Towards more Effective Collaboration Workspaces: From Collaboration Technologies to Patterns (4th Collaboration@Work Experts Group Meeting, Brussels, 18-19 January 2006).

[MRO06] A.I. Molina, M.A. Redondo, and M. Ortega. Applying Pattern-Based Techniques to Design Groupware Applications. Y. Luo (Ed.): CDVE 2006, LNCS 4101, 2006. Springer-Verlag Berlin Heidelberg, pp. 225 – 233.

[Sc07] Doug Schuler et al., Liberating Voices! Pattern Language Project: http://trout.cpsr.org/program/sphere/patterns

[Sch02] Till Schümmer. Constructing a Groupware Pattern Language. Workshop on Socio-Technical Pattern Languages. CSCW 2002 November 16-20 2002. New Orleans, 2002.

[Sch03] Till Schümmer: “Room” in: Schuemmer, T.; Fernandez, A.; Holmer, T.: “The Catalogue of Groupware-Patterns”. Available online at: http://wwwpi6.fernuni-hagen.de:8080/gw-patterns/29 - Darmstadt, 26 November 2003.

[Sch07] Till Schümmer, et al., Patterns-4-Groupware project: http://wwwpi6.fernuni-hagen.de:8080/gw-patterns

[SL07] Till Schümmer and Stephan Lukosch, Patterns for Computer-Mediated Interaction, Wiley & Sons, ISBN: 978-0-470-02561-1, 2007.

[TDC02] John Thomas, Catalina Danis, and Sharon Greene. Socio-Technical Pattern Language Proposal. Workshop on Socio-Technical Pattern Languages. Part of CSCW 2002 November 16-20 2002. New Orleans.

[Th01] John Thomas. “Who Speaks for Wolf?”. Available online at: http://www.truthtable.com/Whospeaksforwolf.html 2001.

[Ti05] Jenifer Tidwell. Designing Interfaces. O’Reilly, ISBN: 0596008031, 2005

[Ya07] Yahoo!, Design Pattern Library: http://com2.devnet.scd.yahoo.com/ypatterns/

[Wa96] Douglas Walton. Argumentation Schemes for Presumptive Reasoning. Mahwah, N. J., Erlbaum, 1996.

[Wa99] Douglas Walton. The new Dialectic: A Method of Evaluating an Argument Used for Some Purpose in a Given Case. ProtoSociology 13 (1999), pp. 70–91.