An Evolving Pattern Library for Collaborative Project Documentation

Von der Fakultät für Mathematik, Informatik und Naturwissenschaften der RWTH Aachen University zur Erlangung des akademischen Grades eines Doktors der Naturwissenschaften genehmigte Dissertation

vorgelegt von

Diplom-Informatiker
René Reiners

aus Mönchengladbach-Rheydt, Deutschland

Berichter: Prof. Dr. Matthias Jarke
Prof. Dr. Uwe Zdun

Tag der mündlichen Prüfung: 11. Dezember 2013

Diese Dissertation ist auf den Internetseiten der Hochschulbibliothek verfügbar.

Abstract

In distributed research projects, the involved personnel acquires lots of technical and domain-specific knowledge. Generalizable outcomes of single project partners that are relevant to all stakeholders need to be distributed within the project. Since the involved parties may possess very different professional backgrounds, specific jargon and different ways to document results may lead to inefficient exchange. Therefore, within this kind of projects, it is a hard task to communicate and keep general project knowledge current. It is likewise difficult to provide achieved results for future projects.

This thesis proposes to collaboratively describe project outcomes and gathered domain knowledge as evolving design patterns. A design pattern must, by definition, be easy to read and understand by non-experts. This enables all stakeholders to understand the described contents without requiring specific background knowledge. The thesis develops a collaborative pattern formulation and validation process that takes into account the special conditions of joint research projects. Therewith, the research and development personnel can easily draft project knowledge in parallel to their efforts as initial design patterns that are refined over time.

From initial ideas or open problems the formulations evolve to validated and reusable patterns that are organized within a dynamically growing pattern library structure. A light-weight role model supports composing new and reviewing existing submissions as well as administrating the library structure. The derived maturation process ensures the formulation quality of the pattern by reflecting the proposals and opinions of all participants. In addition, all contributors collect evidence to support or refute the solutions suggested by a pattern. The research methodology included, besides research in literature, user-centered, iterative design methods that involve representatives of the research and development personnel.

The approach was concretely implemented within a customizable technical platform and substantially validated in an existing distributed research project. A second study was conducted in an academic context in which patterns are used to incrementally document findings. These validations indicate that the evolving pattern library concept is understandable and achieves high acceptance. Furthermore, it can be a useful tool for improving knowledge exchange and accumulation within and across projects.

Zusammenfassung

Innerhalb verteilter Verbundprojekte erwirbt das beteiligte Personal eine große Menge an technischem und domänenspezifischem Wissen. Verallgemeinerbare Ergebnisse einzelner Projektpartner die für alle Interessensvertreter von Bedeutung sind, müssen innerhalb des Projektes zugänglich gemacht werden. Da die beteiligten Parteien zumeist über sehr unterschiedliche fachliche Hintergründe verfügen, können verwendetes Fachvokabular und verschiedene Arten der Ergebnisdokumentation zu ineffizientem Austausch führen. Deshalb ist es mühsam, allgemeines Projektwissen innerhalb dieser Art von Projekten zu kommunizieren und aktuell zu halten. Es ist ebenfalls schwierig, erreichte Ergebnisse für zukünftige Projekte bereitzustellen.

Der Ansatz dieser Arbeit besteht darin, Projektergebnisse sowie gesammelte Erkenntnisse über die Anwendungsdomäne gemeinsam als sich entwickelnde Entwurfsmuster zu beschreiben. Ein Entwurfsmuster muss per Definition für Nicht-Experten einfach verständlich sein. Dies ermöglicht allen Interessensvertretern, die beschriebenen Konzepte ohne Spezialwissen zu verstehen. Die vorliegende Arbeit entwickelt einen gemeinsamen Prozess zur Formulierung und Validierung von Entwurfsmustern, der die besonderen Bedingungen von Verbundprojekten berücksichtigt. Somit kann das Forschungs- und Entwicklungspersonal Projektwissen auf einfache Art parallel zur Projektarbeit als anfängliche Entwurfsmuster skizzieren und über die Zeit verfeinern.

Von anfänglichen Ideen oder offenen Problemen entwickeln sich die Formulierungen zu validierten und wiederverwendbaren Entwurfsmustern, die innerhalb einer dynamisch wachsenden Entwurfsmuster-Bibliothek organisiert werden. Ein einfaches Rollenmodell unterstützt das Verfassen neuer und die Überprüfung bestehender Beiträge sowie die Administration der Bibliotheksstruktur. Der hergeleitete Reifeprozess stellt die Formulierungsqualität der Entwurfsmuster sicher, indem die Vorschläge und Meinungen aller Teilnehmer reflektiert werden. Weiterhin sammeln die Benutzer Beweise, um die innerhalb eines Entwurfsmusters vorgeschlagene Lösung zu unterstützen oder zu widerlegen. Die durchgeführten Forschungsarbeiten zur Erlangung der präsentierten Lösung beruhen, neben einer Literaturrecherche, auf einer nutzerzentrierten und iterativen Entwicklungsmethodik, die Vertreter des Forschungs- und Entwicklungspersonals miteinbezieht.

Der Ansatz wurde konkret mit Hilfe einer anpassbaren technischen Plattform realisiert und in einer umfangreichen Studie im Rahmen eines existierenden verteilten Verbundprojektes validiert. Eine zweite Studie wurde in einem akademischen Kontext durchgeführt, in dem Entwurfsmuster dazu verwendet wurden, Ergebnisse schrittweise zu dokumentieren. Diese Validierungen zeigen, dass das Konzept einer sich entwickelnden Entwurfsmusterbibliothek verständlich ist und hohe Akzeptanz findet. Darüber hinaus stellt es ein praktisch nutzbares Werkzeug dar, um den Wissenstransfer und -aufbau innerhalb und zwischen Projekten zu verbessern.

Acknowledgements

As a matter of course, this dissertation was compiled solely. However, the derivation of its topic and the engineered results were performed similarly to the described pattern formulation process, i.e. based on collaboration, iterative refinement and incorporation of feedback from discussions.

First of all, I would like to thank my advisor, Prof. Dr. Matthias Jarke, for accompanying my research work, especially in the final phase of this thesis.

Thanks also go to Prof. Dr. Uwe Zdun for taking over the role as co-advisor and the helpful discussions throughout the PLoP 2012 and EuroPLoP 2013 conferences.

My colleagues at the Fraunhofer Institute for Applied Information Technology FIT continuously helped me during my research efforts to keep motivation and confidence. In this regard, great thanks go to the whole *User-Centered Ubiquitous Computing Group*. All of you were of great support and represent a team one may often only hope for.

I would like to emphasize my gratitude towards the head of our group, Dr. Markus Eisenhauer, who constantly supported me by making it possible to integrate my dissertation into our daily work. Thanks also go to Dr. Andreas Zimmermann who helped structuring my thoughts during the focusing phase of the dissertation concept.

A team also changes over time. Therefore, I would like to take the chance to say thank you to our former head of our department, Prof. Dr. Reinhard Oppermann for his support and encouragement that "it will work out". His feedback during presentations and discussions helped to shape the frame for the work as it shows now.

The final phase of this work was written in solitude, meaning that I had to give up an excellent daily work environment from my first day at Fraunhofer FIT. My former colleague, Dr. Christian Prause, who changed position, always was a patient discussion partner and good friend during work and beyond.

The latter is to be said in particular about my colleagues Marco Jahn, Uwe Kirschenmann and Mark Vinkovits who constantly monitored my physical and psychological condition. They always made sure that I remained trained and reduced stress on our workout device situated in our premises.

Not to forget the good spirit of our group: Andrea Bernards who has always been able to talk to me about life besides computer-science and always helped me to survive the organizational demands appearing throughout our daily business.

Beyond our group's borders, Prof. Dr. Wolfgang Prinz always showed interest in exchanging knowledge and supported offered master thesis. Judith Flory provided important feedback from another, non-computer-scientific perspective.

Valuable expert and user feedback was gathered during the workshops and case studies conducted in the scope of the BRIDGE project and the ARL Summer School event. Thanks to all 50 participants of the workshops! I apologize for not listing you all at this point. The same is to be said about the participants and shepherds of the writers' workshops at the PLoP 2012 and EuroPLoP 2013 conferences.

Regarding the BRIDGE project, special thanks go to the supporters of the developed approach from the very first minute, Ragnhild Halvorsrud, Aslak Wegner-Eide, Daniela Pohl, Monika Büscher and Marc Jentsch. Common discussions and publications always were a pleasure. I could rely on their support at any time.

From the common efforts for the ARL workshops, I got to know a very friendly research group at Reutlingen University, Germany. Prof. Dr. Alfred Zimmermann and his team, especially Michael Falkenthal, always provided oportunities for discussion and experimentation. Thanks to their interest, the approach could be further developed.

My family and friends continuously reminded me that there is still life besides work. They showed understanding and gave me support all the time. Special thanks at this point go to my best friend and best man Sebastian Kayser. Despite some distance, he always managed to keep our friendship alive across geographical borders.

My parents, Renate and Heinz-Willy Reiners, never doubted that the day of completing this thesis would come. They always had the energy to support and encourage me, even in situations that demanded their full power. Ursula and Matthias Komischke, my parents-in-law, equally gave me moral strength and always provided an agreeable environment since I know them. Thanks also go to Simone Komischke and Henning Pillekamp who always provided nice distractions by organizing parties or simply changing their accomodation.

Finally, I would like to dedicate this work to the person that had to endure my moods during all the time and never got tired of proof-reading all these texts:

My beloved wife, Beatrice Reiners.

If there is a pattern for becoming happy in life, then you are the proven solution.

René Reiners

in December 2013

Contents

List of Figures

List of Tables

List of Abbreviations

ARL	Architecture Reference Lab
BRIDGE	Project acronym. Full title: ”Bridging resources and agencies in large-scale emergency management”
CORDIS	Community Research and Development Information Service
CP	Collaborative Project
CMS	Content Management System
CRUD	Create, Read, Update, Delete
DAG	Directed Acyclic Graph
DIA	Design, Implement, Analyze
EC	European Commission
EPL	Evolving Pattern Library
ER	Emergency Response
GUI	Graphical User Interface
HCI	Human-Computer-Interaction
IP	Integrated Project
MVC	Model-View-Controller
NoE	Network of Excellence
PL	Pattern Language
PLoP	Pattern Languages of Programs
PMI	Pattern Maturity Indicator
SCE	Safety-Critical Environment
SOA	Service-Oriented Architecture
SVG	Scalable Vector Graphics
UCD	User-Centered Design
UI	User Interface
UML	Unified Modeling Language
WP	Work Package

Chapter 1

Introduction

Distributed research projects involve numerous stakeholders from many different technical and non-technical disciplines such as natural sciences, engineering, psychology, economics or jurisprudence, for example. Depending on the project's focus, target users from specific domains are involved to learn about special demands and conditions, elicit requirements, support design decisions as well as validate the project achievements. Proven solutions are exploited as new products or input for follow-up research projects. In case that the number of project participants grows and institutions are spatially distributed in different countries, activity and information coordination becomes an even more challenging task.

Various disciplines in research projects

Every project partner brings in existing experience, specializations and working practices that need to be aligned with the activities of the other participants. This leads to a mixture of different methods, processes and formats used by each member. Compatibility and applicability issues regarding different kinds of results place burdens for cooperation. In order to better structure the planned activities, work packages are created according to the specializations of the partners and needed efforts for the project vision. This way, experts are able to concentrate on different aspects of the project aims and coordinate their efforts. However, every project member demands for a general overview of the project state in order to align current activities with the overall goal and other work packages. Explored domain knowledge, technological achievements and validated results need to be kept current and communicated to every associate.

Organization of work and responsibilities

Depending on the applied methods of each involved project partner, the media used for communication and documentation strongly varies. Whereas technical oriented work packages usually produce brief technical descriptions accompanied by jargon and specialized diagramming methods, validations produce reports and apply statistical methods for the verification of their results. Domain analysis and exploration produce written reports and recorded media from interviews and observation sessions. Exploitation and demonstration efforts make use of achievements of all other work packages and therefore need to understand the ongoing activities. The different formats in which out-

Different documentation formats

comes are generated are not always suited for the next process step. In order to grasp the essences and gain the overview of existing knowledge, a disproportional amount of time must be invested in relation to the existing project work. In some situations, the time to parse and understand the results cannot be taken and thus, knowledge may be transferred incompletely.

Knowledge exchange in joint research projects

With the increasing number of joint research and development projects, the process of knowledge exchange and mutual learning between members becomes complex due to the heterogeneity of the involved agents (cf. Arranz and de Arroyabe [2009]). A taxonomy for learning and knowledge management for research joint ventures is presented by Revilla et al. [2005]. They correlate the focus regarding existing and newly generated knowledge with the application on structural versus social approaches. In the scope of software engineering, Schneider [2009] presents methods for structuring and reusing knowledge gathered from project participants and throughout the engineering process. Technical outcomes found their ways into project work as collaborative spaces, wikis, and social networks which are set up in private, project-related networks. A study performed by Prause et al. [2010b] regarded applied means of communication. The results show that synchronous as well as asynchronous methods are used ranging from personal meetings, video and phone conferences over to chat, mail and document repositories that are accessed and used on demand.

Jargon and separation remains

The variety of communication and document exchange tools does not avoid the clustering of media and formats used by individual project partners within their work packages. Individual jargon of the specialized work groups is kept. Although this kind of documentation and exchange is extremely important for the project's progress, the contents remain only understandable for experts. Explicit means to distribute gathered knowledge and achievements across the whole project in a way that is understandable for every participant are still missing. Equally, cross-project documentation needs to be prepared in a short and concise format such that it does not take much time to read and understand the contents.

Thesis aim

Therefore, this thesis develops a process for building an incremental knowledge structure that captures and communicates generalizable project results in a format which is understandable for all stakeholders.

1.1 Design Patterns as Means of Communication

Proposal of design patterns

In order to convey generalizable information to the whole project consortium, this work proposes to apply the concept of design patterns for documenting project knowledge. The pattern concept is well-known and is capable of capturing working solutions to recurring problems that a community of experts has developed over time. Patterns originate from the architectural domain where Alexander [1977] presents patterns as readily formulated pieces of solutions for general design problems. Gamma et al. [1994]

transferred the idea to software design. Other areas such as user interface design (cf. Tidwell [2011]), human-computer interaction (cf. Borchers [2001]) or website design (cf. Graham [2003] and van Duyne et al. [2007]) made use the pattern concept to share their knowledge. The solution proposed by a design pattern should be generic rather than specific, such that it can be implemented in numerous different ways. Several other domains besides computer science adopted the concept and therewith formulate patterns for a variety of topics. Prominent examples are presented by Manns and Rising [2005] for organizational processes development, Coplien and Harrison [2005] with regard organizational advice and Bergin et al. [2012] who present pedagogical patterns. The authors usually make use of natural language in order to avoid specific vocabulary and keeping the formulations understandable for non-experts. Recent research takes into account semi-formal and formal approaches in order to automize the structuring, retrieval and selection processes for patterns (cf. Cornils and Hedin [2000], Montero et al. [2005], Pavlič et al. [2009], Smith and Stotts [2002] and Eden et al. [1997]).

Patterns as common vocabulary

The presented approach considers patterns not only as a way to capture and represent design knowledge but also as means of communication and documentation within and beyond the current project. This way, patterns are used to capture domain, process as well as technical knowledge during all phases of the project lifecycle from many perspectives. The aim is to improve the exchange between different disciplines by applying a common vocabulary that gives project members more expressiveness regarding the problem domain. Many existing pattern collections keep formulations readable and easy to understand. This way, patterns can serve as a *lingua franca* between stakeholders and interested new or experienced groups in the field. Erickson [2000] supports this view and addresses the diversity within projects dealing with interaction design. He follows an approach towards meta pattern languages that can be adapted "site-specifically" for the project at hand. A well-known vocabulary in software engineering actually was created by Gamma et al. [1994]. Following their denominations, developers and system architects do not describe the concepts to a solution but directly use the name of the concept that is directly understood by others with expertise in the field.

1.2 Research Questions and Approach

Patterns should reflect the project's progress

Achievements in research projects are usually reached stepwise in an asynchronous manner and refined over time. This progress must be reflected by the current state of a pattern's formulation. Besides existing expertise, new findings and visions should be formulated as patterns as soon as possible in parallel to the project work. In contrary, design patterns are traditionally formulated *after* gathering a lot of experience by experts over time and providing them for upcoming challenges. This means that the concept needs to be extended such that patterns *evolve* gradually with every new achievement and validation. This way, they mature incrementally according the current state of the project outcome they describe. Requirements represent a starting point for formulating project-relevant patterns since they place demands on the development of concepts, systems and applications. Results and feedback gathered from prototypes, evaluations and validations need to be reflected by continuously updating existing patterns. Finally, exploitation benefits from patterns since they represent a condensed view on specific aspects. The repository in which all patterns are structured likewise needs to be refined continuously. Responsibilities for formulating and validating the pattern must be indicated and assigned but kept lightweight enough to remain easy to understand. Rules for the pattern maturation process need to be derived and can be adapted to the scope of the current research project. In the end, the derived patterns contain project-wide knowledge for follow-up research and future product development.

Research questions

The proposed extension of the pattern concept and its mechanisms for a pattern's formulation and validation process place new demands and lead to the following research questions that are treated in the scope of this thesis:

- What qualities are missing in currently existing pattern mining and formulation processes that place new requirements for collaboratively formulating and validating design patterns according to the current state of project results?
- How should an incremental pattern formulation and validation process be structured in order to reflect a pattern's reliability?
- How can a pattern's formulation quality be ensured and validity be measured?
- Which activities are performed by which roles such that the process remains easy to apply and to understand by the users?
- What kind of a technical platform is suited to realize the developed approach and what are accepted ways of showing progress information as well as activity?

1.3 Methodology

Iterative, user-centered design

In order to tackle the addressed challenges, a user-centered, iterative design and engineering approach was applied as introduced by Nielsen [1993] who suggests to directly involve users in a user-centered design (UCD) philosophy. He describes development cycles consisting of the phases "design", "implement" and "analyze" as illustrated in Figure 1.1. Iterative design is a well-suited instrument for system development that focuses on early evaluations and prototyping design ideas together with the target users. At the beginning of the development phase, storyboards or paper prototypes are created and evaluated. These methods help to gain initial feedback about the envisioned concepts. Over time, realizations become more sophisticated until a final prototype serves as foundation for the productive implementation.

User involvement

Hansen [1971] states that it is important for designers and developers to clearly know about the users' special demands and behavior and what kinds of tasks are performed

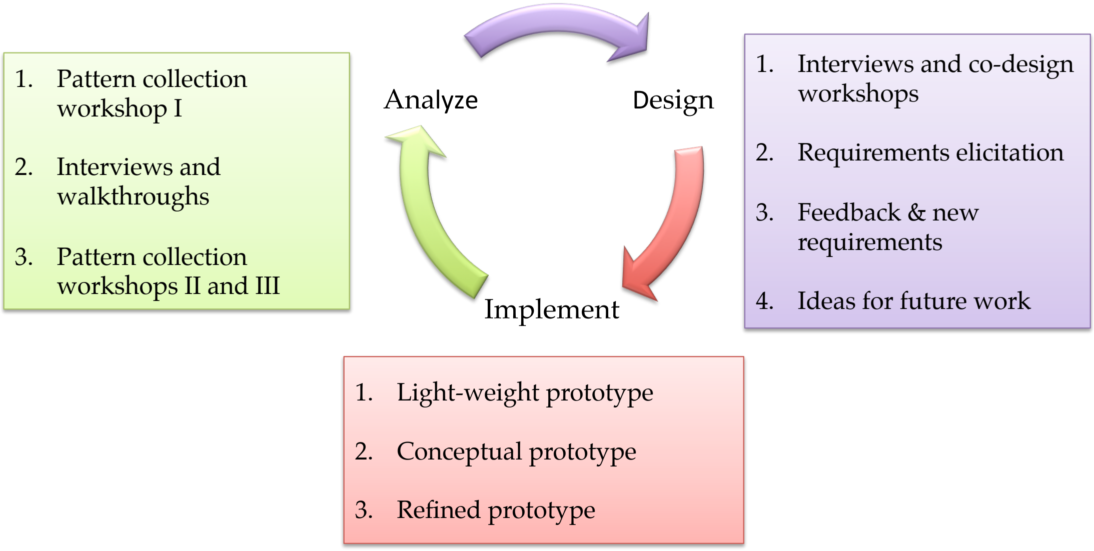


Figure 1.1: Iterative design cycles allow to continuously validate and refine implementations.

by them. Therefore, representatives from research and development were continuously involved in the engineering efforts and assisted in modifying the approach till its technical realization and validation in real project settings. Muller et al. [1993] present a taxonomy of methods and techniques at different times of development in correlation with the mainly involved user groups reaching from designers to end users. The taxonomy also takes into account the group sizes of needed for each approach and motivated the methods used in the scope of this work.

Interviews and co-design workshops

As starting point for the engineering efforts, semi-structured interviews were held with research and development personnel (cf. Wood [1996]). The aim was to gain a better understanding of existing problems regarding project-wide documentation and knowledge exchange. Repeated co-design workshops helped to collect opinions and understand reasons of the working practices of the addressed stakeholders (cf. Anderson and Crocca [1992]). The workshops were later used to elicit explicit requirements for the envisioned concept after collecting goals and expectations and encountered problems from all participants. In combination with the future workshops method as described by Floyd et al. [1989], the identification of missing qualities of current approaches led to the derivation of a collaborative pattern formulation process that takes into account the assurance of formulation quality as well as the validity of made suggestions for solutions. Cooperative prototyping sessions, as introduced by Beyer and Holtzblatt [1998], discussed and adapted possible processes, needed roles and rules. Therefore, the presented solution suites the needs of the research and development personnel while remaining easy to handle without causing too much overhead.

Parallel design yields better results

At each workshop session, different alternatives of the presented concepts were presented and discussed as inspired by Nielsen and Fabe [1996] and Dow et al. [2010]. They show that parallel prototypes result in better designs based on more diversity and more task-specific confidence. During their studies, designers stated that their self-efficacy as a measure of task-oriented confidence increased during parallel design sessions. The evaluators do no longer fear to destroy the complete work of the design but to discuss the pros and cons about all the ideas presented (cf. Tohidi et al. [2006]). The refinement of only one prototype in a serial process, on the contrary, may hinder the assessment of other alternatives and even narrow the possible design space (cf. Buxton [2007]). Design alternatives encourage participants to introduce new ideas.

Implementation for workshops and final validation

In parallel to the conceptional efforts, a practical prototype was implemented and refined over two iterations in the scope of the joint research project BRIDGE[1]. The implementation supported discussions on the concepts and helped to discover remaining difficulties and missing features. The final pattern formulation and maturation process to be used during the lifetime of a research project was validated within an extensive study with project participants. It asked for the formulation of new patterns as well as reviewing and validating existing ones. As further validation of the concept and its implementation, a short-term study in the domain of service-oriented architectures where pattern repositories are used for learning and teaching purposes was conducted at Reutlingen University, Germany. Both studies show the potential of the approach to be embedded in the workflow activities under the open conditions of research projects.

1.4 Contributions

Patterns as micro-documentations

The contributions of this thesis cover three aspects as illustrated in Figure 1.2. The first aspect describes an iterative collaborative process for mining, formulating and validating patterns. For the approach presented in this work, the pattern language concept is adapted in order to be applied as *evolving micro-documentation* throughout the engineering and knowledge management process. Instead of formulating approved and working solutions resulting from long-term experience, every finding and idea is formulated as an, initially incomplete, pattern idea which matures in parallel to the project lifetime and beyond. The developed concept aims at enabling *all* project participants to quickly formulate, review, comment and refine existing pieces of knowledge and to find supporting or refuting evidence regarding the described concept.

Technical infrastructure

The second aspect focuses on the development of a technical pattern library prototype that is adaptable to the individual project environment. A low learning-curve allows all participants to easily browse, contribute, and improve the pattern formulations. Means of visualizing activities of all participants, new contributions, as well as the library structure and pattern maturity are provided. Additionally, the platform serves as vali-

[1] www.bridgeproject.eu

dation testbed for the presented pattern formulation and validation process by extending an existing content management system.

Formulated design patterns

The third aspect describes the results of the application of the implemented approach in a real project setting. The collection of design patterns formulated during the engineering efforts within the BRIDGE project supports the presented concepts of this thesis. It contains patterns in different maturity states that were formulated throughout conceptional phases and validated in parallel to the project results. The presented collection reflects the engineering activities that were performed during the project's lifetime and is considered as seeding set of patterns to motivate future contributions.

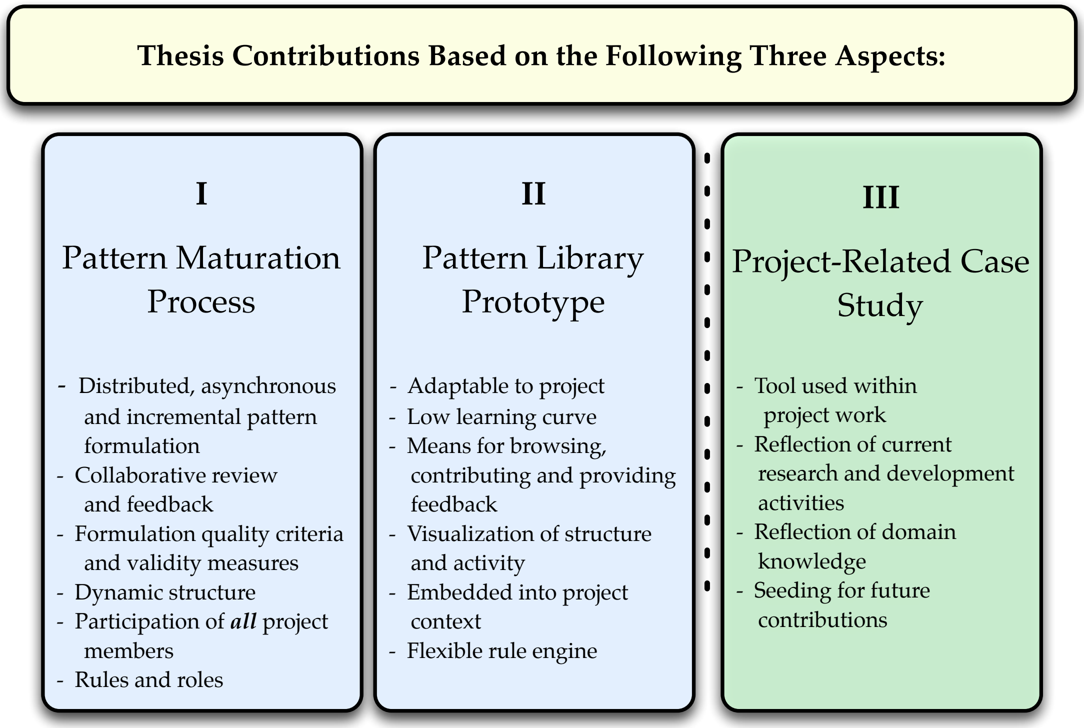


Figure 1.2: Overview of the three contributions of this thesis.

1.5 Thesis Structure

The following Chapter 2 presents the concept of design patterns, its origins and different applications in technical as well as non-technical domains. Together with approaches of organizing patterns in collections and pattern languages the chapter presents a formal definition of pattern languages given by Borchers [2001]. A discussion on pattern mining and formulation approaches founds the basis of the discussion of needed extensions for achieving the aims of this thesis.

Chapter 3 explains in more detail the general structure of large-scale, highly distributed research projects and their characteristics as well as their dynamics based on personal experiences that were gathered during the project work at Fraunhofer FIT over the last five years. From a general point of view, the chapter treats the problems concerning communication and knowledge exchange between different work packages.

The subsequent Chapter 4 shows the research agenda for the iterative engineering approach that was applied to derive the concepts presented in this thesis. The first conception of the collaborative pattern formulation process is based on the problem analysis that was explored during user workshops and considerations on past project experiences. Based on identified shortcomings of the situation and existing approaches, respectively, requirements are formulated that are based on problem analysis workshops with expert groups from the project domains.

Chapter 5 presents evolving pattern library (EPL) concept together with a process model that introduces pattern maturity states taking into account a pattern's formulation quality and evidence for its validation. A light-weight role model allows for organizing responsibilities regarding active contributions, assessment as well as managing the structural development of the library. The traditional pattern structures are extended to fit into the presented approach.

For validating the EPL concept a technical platform for was implemented in a user-centered approach as described by Chapter 6. Means for browsing and navigating the patterns are presented together with functionalities to contribute, rate and discuss them. Mechanisms for showing the community's activities and therefore the liveliness of the whole portal are explained. The description starts with a first prototypical implementation that serves as foundation for discussions on requirements and demanded features. At the end of the chapter, a practical usage scenario shows the acceptance of the approach together with its practical realization that outlines issues for future work. The findings are based on technical and survey-based validations and include qualitative feedback from the participants.

The practical usage of the developed evolving pattern library concept in the existing research project BRIDGE are presented in Chapter 7. Besides the practical results in the form of formulated and partially validated patterns, a substantial survey shows the acceptance and understandability of the approach and its implementation. As beneficial outcome of the engineering work put into the problem and requirements analysis as well as the design and validation of the evolving pattern library concept, patterns gathered in the domain of emergency response are assembled in Appendix E.

Chapter 8 summarizes the applied methods for engineering the derived concepts and achieved results. Considerations on future challenges and proposals for extending the derived collaborative pattern formulation approach shape in the second part of the chapter before personal remarks conclude this thesis.

Chapter 2

Background on Design Patterns

Recurring design problems are a well-known phenomenon affecting many different domains. Not only in the technical sector but also in design areas like architecture, handcraft, construction or interface design, people see themselves faced with problems they do not directly know a solution for. However, it is possible that the same or a similar problem has also been treated by someone else and that a solution or guideline was found. In a best case scenario, this "best practice" was shared with others. Examples for distribution channels are oral dissemination, demonstrations while working on problem as well as books and guidelines that are used for education or counsel.

Best practices for recurring problems

Design patterns are a well-known approach of formulating pieces of knowledge that goes beyond the propagation of best practices or recommendations for operations. Originally, design patterns are formulated in a way that non-experts are able to understand the context in which the problem occurs and the proposed solution. This chapter introduces to the concept of design patterns and its application in different domains.

Design patterns explain solutions

In a second step, ways to cluster and structure sets of design patterns in collections and pattern languages are presented. Exemplary overviews of pattern collections and languages, respectively are given in each corresponding part. The intention of the samples is to show the variety of pattern language approaches used for documenting knowledge in different levels of technical detail and different domains. An extensive amount of formulated patterns exists such that only a rough survey can be given in the scope of this work to introduce the topic. Due to work environments of the author and study group members, most examples are taken with regard to software engineering and human-computer interaction.

Organization of design patterns

An overview of different *pattern mining* approaches shows different workshop- and process-based methods to discover, formulate and validate patterns. The methods vary in the number of participants, their spatial distribution as well as the embedment into project workflows.

Pattern mining

2.1 Design Patterns

Architecture as origin for design patterns

The concept of design patterns originates as an idea for sharing knowledge and experience and was introduced by the architect Christopher Alexander. He describes successful guidelines for architectural designs as single portions of knowledge and organizes his collection by the spatial dimension starting with landscapes reaching down to quarters, houses and single rooms.

Early pattern definition

Early formulations of design patterns can be found in Christopher Alexander's work *Notes on the Synthesis of Form* (cf. Alexander [1964]). In a later publication, he defines a pattern in a more mathematical way:

Definition 2.1: Design Pattern
The pattern statement is itself broken down into two further parts, an IF part, and a THEN part.
IF: X THEN: Z / PROBLEM: Y
- x defines a set of conditions.
- y defines some problem which is always likely to occur under the conditions x.
- z defines some abstract spatial relation which needs to be present under the conditions x, in order to solve problem y.

Alexander et al. [1968]

Morphological laws from architecture domain

Later on, Alexander gave up this strict definition and captured experience in morphological laws that explain how to design an artifact depending on a specific context such that a particular design problem can be solved. In his books "The Timeless Way of Building" (cf. Alexander [1979]) and "A Pattern Language: Towns, Buildings, Construction" (cf. Alexander [1977]), he collects knowledge from the architecture domain arranged in 253 units of information, i.e., design patterns, in different levels of detail. Besides the aim of relay design knowledge, Alexander intends to reach a wide audience. Each pattern is presented in such a way that also non-domain experts are able to understand the described solution and apply it to their current problem.

2.1.1 Informal Pattern Formulation

Patterns as abstract recurring solutions

Alexander states that each implementation of the same pattern differs in its characteristics even in case they share the same origin. Patterns shall be seen as description of a problem solution as starting-point and not as fixed design rules. They describe and preserve solutions to recurring design problems. It is important to formulate a pattern abstract from a specific solution and thus make it generalizable and transferable to other problems of similar kind. Only relevant problems that are particularly interesting are candidates for pattern formulation, i.e. trivia do not need to be documented.

A design pattern describes the *context* in which it can be applied, the *problem* it addresses and the proposed *solution*. Each design pattern should have a clear and descriptive *name* so that others can easily identify it, and ideally provide an *illustration* that helps to understand the solution. The latter plays an important role. However, the solution is only depicted as a diagram in order not to anticipate a real world design suggestion. This way, it is still kept on an abstract level and can be implemented individually.

Alexandrian pattern structure

Alexander describes *forces* that represent conflicting elements within a problem scope. In the domain of architecture, these can, e.g., be influences on the natural behavior of humans like the unconscious tendency to move towards lighter places within a room. An opposing force is the, usually existing, demand for the availability of seats in a room. One possible solution for these conflicts is to provide seats distributed within the room but also next to windows. With this solution, a compromise for many individuals and situations is available.

Conflicting forces

A *diagram* depicts the intended solution as well as a *summary* about key elements of the presented solution. The following quote from Christopher Alexander gives a basic rule for creating patterns for a specific pair of problem and solution:

Combination of illustrations and text

> ”If you can’t draw a diagram of it, it isn’t a pattern”,
> Alexander [1977], p. 267.

At the end of a pattern, *references to related patterns* present insight into more design details that sometimes can be combined or even contradict each other. It is also possible that the references points towards preceding patterns in order to show the conceptual connection between them.

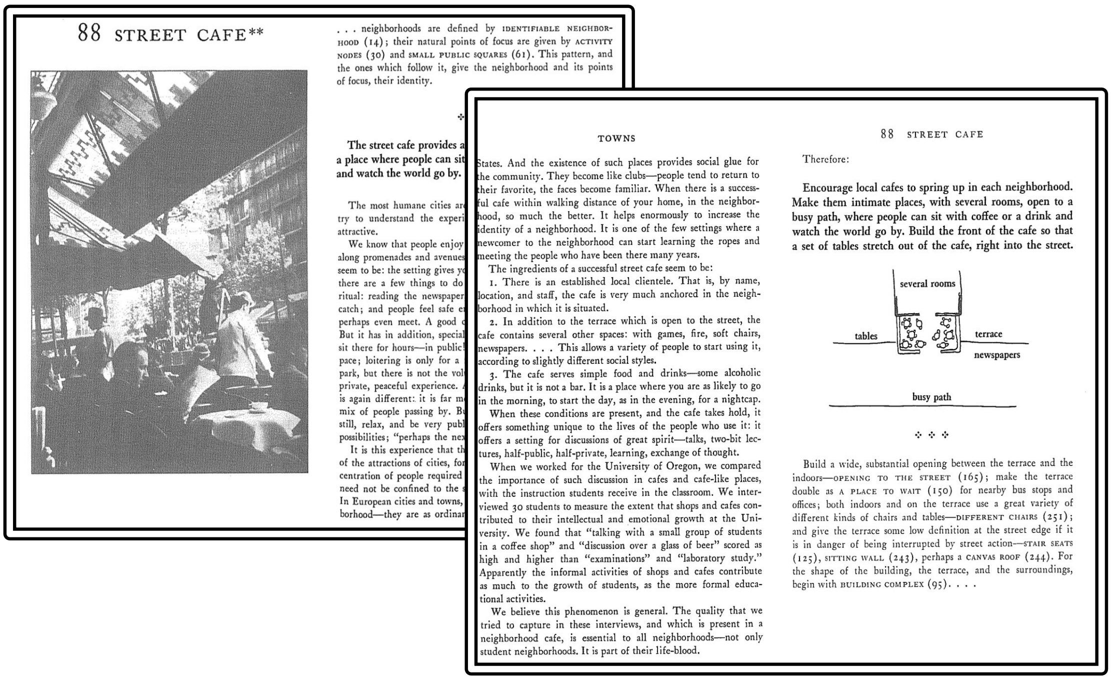
Relations to other patterns

The pattern structure is given implicitly through typesetting. This way, repeating labels do not disturb the natural description and read-flow of the pattern. The solution, for example, is always marked by bold face letters starting with ”Therefore,...”’ which makes it easy to skip to that part directly or to go through several parts within the pattern description.

Differentiation by typesetting

Figure 2.1 shows an example of an architectural design pattern called ”Street Cafe” that describes the qualities of these special places within cities where people can relax, observe and be seen by others. The Street Cafe pattern is embedded in the context of ”Neighborhoods”, ”Nodes for Activities” and ”Public Squares” that each are described by different patterns. After introducing the reasons for demands for these kinds of places, a summary of the pattern is given, then the qualities and required features are explained in detail. In conjunction with a summarizing design advice, an illustration shows in general the arrangement of a street cafe. In the end, references to more details are given on patterns that discuss more details like more purposes of a cafe (”Places to Wait”) or the raising the attractiveness for customers (”Opening to the Street”). The whole pattern is formulated in natural language that abstains from domain-specific jargon so that the text remains easy to understand for non-domain experts.

Street Cafe example


88 STREET CAFE**

. . . neighborhoods are defined by IDENTIFIABLE NEIGHBORHOOD (14); their natural points of focus are given by ACTIVITY NODES (30) and SMALL PUBLIC SQUARES (61). This pattern, and the ones which follow it, give the neighborhood and its points of focus, their identity.

✣

The street cafe provides a
a place where people can sit
and watch the world go by.

The most humane cities are
try to understand the experi
attractive.
We know that people enjoy
along promenades and avenues
seem to be: the setting gives yo
there are a few things to do
ritual: reading the newspaper
catch; and people feel safe e
perhaps even meet. A good c
But it has in addition, special
sit there for hours—in public!
pace; loitering is only for a
park, but there is not the vol
private, peaceful experience. A
is again different: it is far m
mix of people passing by. Bu
still, relax, and be very publ
possibilities; "perhaps the nex
It is this experience that th
of the attractions of cities, for
centration of people required
need not be confined to the s
In European cities and towns,
borhood—they are as ordinar

TOWNS

States. And the existence of such places provides social glue for the community. They become like clubs—people tend to return to their favorite, the faces become familiar. When there is a successful cafe within walking distance of your home, in the neighborhood, so much the better. It helps enormously to increase the identity of a neighborhood. It is one of the few settings where a newcomer to the neighborhood can start learning the ropes and meeting the people who have been there many years.

The ingredients of a successful street cafe seem to be:

1. There is an established local clientele. That is, by name, location, and staff, the cafe is very much anchored in the neighborhood in which it is situated.

2. In addition to the terrace which is open to the street, the cafe contains several other spaces: with games, fire, soft chairs, newspapers. . . . This allows a variety of people to start using it, according to slightly different social styles.

3. The cafe serves simple food and drinks—some alcoholic drinks, but it is not a bar. It is a place where you are as likely to go in the morning, to start the day, as in the evening, for a nightcap.

When these conditions are present, and the cafe takes hold, it offers something unique to the lives of the people who use it: it offers a setting for discussions of great spirit—talks, two-bit lectures, half-public, half-private, learning, exchange of thought.

When we worked for the University of Oregon, we compared the importance of such discussion in cafes and cafe-like places, with the instruction students receive in the classroom. We interviewed 30 students to measure the extent that shops and cafes contributed to their intellectual and emotional growth at the University. We found that "talking with a small group of students in a coffee shop" and "discussion over a glass of beer" scored as high and higher than "examinations" and "laboratory study." Apparently the informal activities of shops and cafes contribute as much to the growth of students, as the more formal educational activities.

We believe this phenomenon is general. The quality that we tried to capture in these interviews, and which is present in a neighborhood cafe, is essential to all neighborhoods—not only student neighborhoods. It is part of their life-blood.

88 STREET CAFE

Therefore:

Encourage local cafes to spring up in each neighborhood. Make them intimate places, with several rooms, open to a busy path, where people can sit with coffee or a drink and watch the world go by. Build the front of the cafe so that a set of tables stretch out of the cafe, right into the street.

✣ ✣ ✣

Build a wide, substantial opening between the terrace and the indoors—OPENING TO THE STREET (165); make the terrace double as A PLACE TO WAIT (150) for nearby bus stops and offices; both indoors and on the terrace use a great variety of different kinds of chairs and tables—DIFFERENT CHAIRS (251); and give the terrace some low definition at the street edge if it is in danger of being interrupted by street action—STAIR SEATS (125), SITTING WALL (243), perhaps a CANVAS ROOF (244). For the shape of the building, the terrace, and the surroundings, begin with BUILDING COMPLEX (95). . . .

Figure 2.1: Extract of the "Street Cafe" pattern example taken from Alexander [1977], pp. 436–439.

One format for many stakeholders

With this way of structuring expressing design knowledge, Alexander presents a format to address different stakeholders in a project such as:

- Architects with long term experience from working in the field but willing to acquire existing knowledge.
- Students who begin to learn in the field and who are interested in discussing and thinking about existing solutions and concepts.
- "End users", i.e., customers who want to understand and learn about design practices without the need to learn domain-specific expressions and knowledge.

Inspiration for other domains

The concept of design patterns was picked up by many other domains for capturing knowledge in software-engineering and non-technical fields. Bayle et al. [1998], e.g., formulate "activity patterns" not only as means of sharing knowledge about working solutions but also to describe human behavior and contextual influences. Kruschitz and Hitz [2010] show that design patterns are considered by a majority to support the engineering and design work in the domain of human-computer-interaction design. They are often applied, reused and regarded as useful. Beck et al. [1996] show the relevance for design patterns in industrial design processes by describing their experience within six different industrial teams. They point out the improvement on communication and capturing design essentials. At the same time they state that patterns do not necessarily need to be object-oriented. This hint already underlines the potential for general applicability of the design pattern concept. On the contrary, the authors state that they experienced that developers and designers can better benefit from patterns if they are themselves involved in a pattern writing process. The creation of generally applicable patterns, however, is considered as a difficult task.

Required knowledge

The degree of knowledge that readers require in order to start working with the patterns is also different across pattern languages. Gamma et al. [1994] specifically address software developers with their technical description and pseudo-code example in combination with technical diagrams. Other authors like Borchers [2001], Coplien and Harrison [2005], Schümmer and Lukosch [2007] or Tidwell [2011] make use of less technical language and provide introductory patterns that explain the different problem fields in more details (cf. Section 2.2.2). From the latter approaches, novices in the field are given a starting point to learn about the topic. Specialized and compact solutions to problems can still be selected by experts in the field.

Different readership

van Welie and Traetteberg [2000] differentiate between the *end-user's* and *designer's* perspective for formulating and reading interface design patterns. Whereas designers are usually able to understand the rationale behind patterns from both perspectives, users generally ask for usability benefits. The authors also argue that it is easy to identify any kind of patterns from user interfaces. However, it is difficult to find patterns that really improve the usability of an application. For different reasons, like placing advertisement or cross-links, design patterns focusing on enterprise-related interests are patterns out of the authors' scope of interest.

Problem categories for patterns

In later work, van Welie distinguishes between different problem classes that users normally struggle with. The latter supports users in case they decide on changing the way of working with the currently used object or application. The differentiation is based on the classification given by Norman [1988] and described as follows:

- *Visibility* problems that make it hard for users to understand how to use the device or application.
- *Affordances* concerning the perceived and actual properties of an object that suggest certain kinds of usages.
- *Natural mappings* that create clear relationships between users' goals and the provided mechanisms.
- *Constraints* that reduce the number of actions that users can perform.
- *Conceptual models* which should make it easy to understand how the internal mechanism of an object or application work.
- Comprehensible *feedback* mechanisms about the users actions.

Degree of usability

Concerning the rationale of a design pattern, i.e., assure its validity, van Welie applies measures of *performance speed*, *learnability*, *memorability*, *satisfaction*, *time for task completion* and *number of errors*.

2.1.2 (Semi-) Formal Pattern Formulation

Formalism provides higher precision

Besides the described informal way of formulating and discussing design solutions, semi-formal and formal approaches were developed. Although informal representation helps users to understand the rationale behind a design pattern, there is a limitation to precision due to the use of natural language and the semantic ambiguity (cf. Montero et al. [2005]). This disadvantage does not allow for any level of automation to resolve widely recognized problems such as finding and selecting the appropriate design patterns or applying them (cf. Rosengard and Ursu [2004]). The following paragraphs briefly discuss this different way of formulating design patterns.

Better understanding of pattern orchestration

One goal of formalization is to provide a better understanding of design patterns and their composition which helps to know when and how to use patterns properly in order to take advantage of them. In order to resolve issues regarding relationships between design patterns, it is not only important which design patterns are used to solve a given problem, but also in which order they are applied.

UML as semi-formalism

Several approaches aim at representing design patterns in a semi formal way. Most of these attempts are based on the Unified Modeling Language (UML)[1] descriptions (cf. Fontoura and Lucena [2001], Kim et al. [2003], Rosengard and Ursu [2004], Sunyé et al. [2000]). UML helps to specify the structural and behavioral aspects of design patterns using, e.g., class, sequence or collaboration diagrams. However, UML does not convey the intention of a design pattern and consequences of its application which is regarded as important by Rosengard and Ursu [2004].

Research efforts with regard to automating pattern usage

Formal approaches are mostly based on mathematical logic or ontologies as described by Pavlič et al. [2009]. Research activities include semantic search for design patterns, automatic code generation and formal validation of design patterns. Smith and Stotts [2002] extend the sigma calculus, which defines relationships between the elements of object-oriented languages to describe design patterns. Cornils and Hedin [2000] formulate design patterns by using reference attributed grammars with syntactic and context-sensitive rules. Eden et al. [1997] make use of higher-order logic to represent design patterns as logic formulas, which consist of the elements of object-oriented language, i.e., classes, methods or hierarchies, and relationships between them. In a similar manner, Taibi [2003] specifies the structural and behavioral aspects of design patterns using first order and temporal logic, respectively. He distinguished between classes, attributes, methods, objects and untyped values.

Semantics with ontologies

In analogy to the Semantic Web[2] approach, Rosengard and Ursu [2004] introduce the idea of representing patterns as ontologies with a view to the development of tools for "the automatic organization, retrieval and explanation of reusable solutions to software development, codes of good practice and company policies". With the help of ontologies design patterns can be understood by both, humans and machines.

[1]http://uml.org
[2]http://semanticweb.org

Sedlmayr [2008] describes a semi formal approach for work process patterns for patient treatments. Usually, guidelines for handling specific situations during diagnostics and treatments are known to a large degree by physicians. However, these workflow recommendations are available as collection in books that need to be consulted when needed. Based on the workflow patterns and its control structures provided by van der Aalst et al. [2003], Sedlmayr defines processes by using a workflow modeling language based on Java. The domain-based knowledge is defined within an ontology. Computer-based support is then given by linking model to the current patient's health. Based on the data and modeled workflow rules automated, goal-based suggestion support is provided that enables physicians to quickly react to current situations. Still, the last decision is made by human experts. The system exclusively provides decision aids. In this approach, patterns are understood as goal-driven sequences of actions to undertake in specific situations. This interpretation is stricter than the one explained in Section 2.1.1 where design patterns are used for explanation, teaching and communication.

Medical decision support

Semi-formal and formal formulation approaches for describing design patterns are hard to understand by readers who lack the needed skills and therefore cannot understand and apply the pattern. This circumstance is a big disadvantage with regard to the approach of using patterns as communication medium (cf. Section 1.4). The shown approaches are mostly focus on the solution part of a pattern. However, in the approach followed in this work, the rationale behind the solution and its trade-offs are important descriptions for readers. These parts contain important information about the application of the solution within a problem context. For the sake of using patterns for documenting developments for many stakeholders within a project, the approach of formulating patterns in an *informal* way will be kept for the approach of this thesis.

Informal format is used for approach

2.2 Design Pattern Organization

The following sections discuss the organization of patterns as collections that group them based on common properties and the stricter form of pattern languages that introduce structures and relations. The latter provide a context for each single pattern and optionally connect one solution to related patterns that treat further concepts in more details or that present an alternative to the current suggestion. The formal definition of a pattern language given by Borchers [2001] is shown alongside with examples for pattern languages in technical and non-technical domains.

Collections and languages

2.2.1 Pattern Collections

A very prominent example of a successful pattern collection in the domain of software engineering regarding questions on issues in object-oriented programming designs was formulated by Gamma et al. [1994]. Their work provides a common vocabulary as a

Clustering by purpose

collection of 23 patterns for software designers. References to already known design concepts that are named inside the pattern language help to exchange about design trade-off and alternatives without the need to explain the concepts behind the names. The authors structure their pattern collection within three main pattern clusters: *behavioral*, *structural* and *creational*. The first cluster deals with the ways of instantiating objects and managing their lifetime. The second one covers the composition of different classes or objects and the third cluster describes the ways in which classes or objects interact and handle responsibility. In contrast to Alexander's approach, the patterns are written in a more technical style. UML diagrams and examples written in pseudo-code illustrate the solution.

Gestural interfaces

Saffer [2008] presents a pattern collection about gestural interfaces in which the patterns are grouped by topics around gestures for touch screens and interactive surfaces and free-form interactive gestures alongside with descriptive overviews about prototyping, documentation and communicating interaction concepts. The patterns are not related among each other so that they independently describe a solution particular problem without a larger context. Examples are given at the end of each pattern that show the application of the described principle in a product or prototype.

Organization along problem classes

Scott and Neil [2009] organize their collection of 75 patterns about interaction design on websites by six design principles concerning direct interaction styles, light-weight contextual tool support, methods to present content within a page, providing invitations to visitors, the use of translations and immediate system reaction. Within the design principles, up to four subcategories further structure the pattern collection. There are no associations between the patterns keeping the collection uncoupled. Readers need to orient by the categorization. The pattern collection represents a distillation of around 30 years of experience gathered by the authors. Besides the discussion of the presented solution, a list of alternatives gives further support. Irregularly, anti-patterns are discussed as inserted paragraphs while discussing the proposed solution.

Mobile application design

Additional examples especially for design patterns of mobile devices can be found within the *Flaminco* design pattern library of Nilsson [2009]. The patterns are not interconnected but handle specific topics concerning interaction and navigation on small screens. Aesthetics as well as suggestions to avoid additional input devices like styluses are interesting patterns relevant for device, interaction and application design.

2.2.2 Pattern Languages

Hierarchies and Interconnections

Alexander [1977] introduces the notion of interconnected patterns as a pattern language (PL) by describing architectural concepts with the aim of formulating the contributions in a generally understandable way, i.e., also for non-domain experts. He organizes his patterns in a hierarchy based on the spatial of the aspects described by the individual pattern. This way, he provides a navigable information structure allowing the reader to start searching for solutions at any point in the language depending on the reader's current problem context.

From landscapes via towns to buildings

Alexander starts with large-scale contexts like landscapes and describes very generally different qualities of possible grounds for towns. Then, different parts of a city are described like residential, commercial and industrial areas. Aspects on public and private transportation as well as important institutions (schools, hospitals, responder facilities or shops) are discussed. This way, the degree of detail is continuously increased until design support is given for houses, gardens up to single rooms. This way, the presented patterns build on top of each other. In order to understand underlying concepts, preceding patterns need to be understood first. Implicitly, the associations between the patterns are mostly pointing at patterns concerned with more details and refinements.

Consideration of alternatives

Multiple associations to other patterns do not demand from the reader to decide for one specific pattern, but suggest considering different design alternatives or a combination of them. However, Alexander neither aims at connecting all patterns nor strictly linking them bidirectionally. He does not focus on a fully interconnected pattern language but to use it as didactic medium for human readers that familiarize themselves with the whole structure. Readers need to mentally add the missing connections.

A Formal Definition for Pattern Languages

Formalism

Borchers [2001] formalizes the structural hierarchy of a pattern language in a stricter way as a *directed acyclic graph (DAG)* with an arbitrary number of incoming and outgoing edges that represent a pattern's context and its references to other patterns:

DEFINITION 2.2: PATTERN LANGUAGE

1. A *pattern language* is a directed acyclic graph $PL = (\wp, \Re)$ with nodes $\wp = \{P_1, ..., P_n\}$ and edges $\Re = \{R_1, ..., R_m\}$.
2. Each node $P \in \wp$ represents a *pattern*.
3. For two nodes $P, Q \in \wp$, we say that P *references* Q if and only if there is a directed edge $R \in \Re$ leading from P to Q.
4. The set of edges pointing away from a node $P \in \wp$ is called *references*, and the set of edges pointing to it is called *context*.
5. Each node $P \in \wp$ is itself a set $P = \{n, r, i, p, f_1...f_i, e_1...e_j, s, d\}$ of a name n, ranking r, illustration i, problem p, with forces $f_1...f_i$, examples $e_1...e_j$, the solution s, and diagram d.

From general to concrete concepts

Based on the proposed definition, Borchers [2001] defines three pattern languages in the scope of publicly accessible interactive music exhibits. From his point of view, especially patterns in human-computer interaction (HCI) need to bridge the gap between users with conceptual knowledge, domain experts and software engineers who are deeply involved in the technical development. The hierarchy within the presented pattern languages reveals more detail knowledge as needed during the design process. From a general concept that is explained in the top levels of the hierarchy, more detailed problems are derived revealing more insight towards concepts of solutions. In order to understand the patterns from the lower levels within the hierarchy, the predecessors must be known by the reader. Novices should start exploring the pattern language from the root element of the graph structure and follow the edges to subsequent patterns. This way, the graph is traversed via a breadth-first search. Figure 2.2 shows As the *World Beat* pattern language as an example for a pattern language that explains the concepts of Blues music. The derived patterns stepwise reveal conceptual details about harmony, melody and rhythm on which Blues-styles build. The formulated knowledge served as starting point for the design of interactive exhibits.

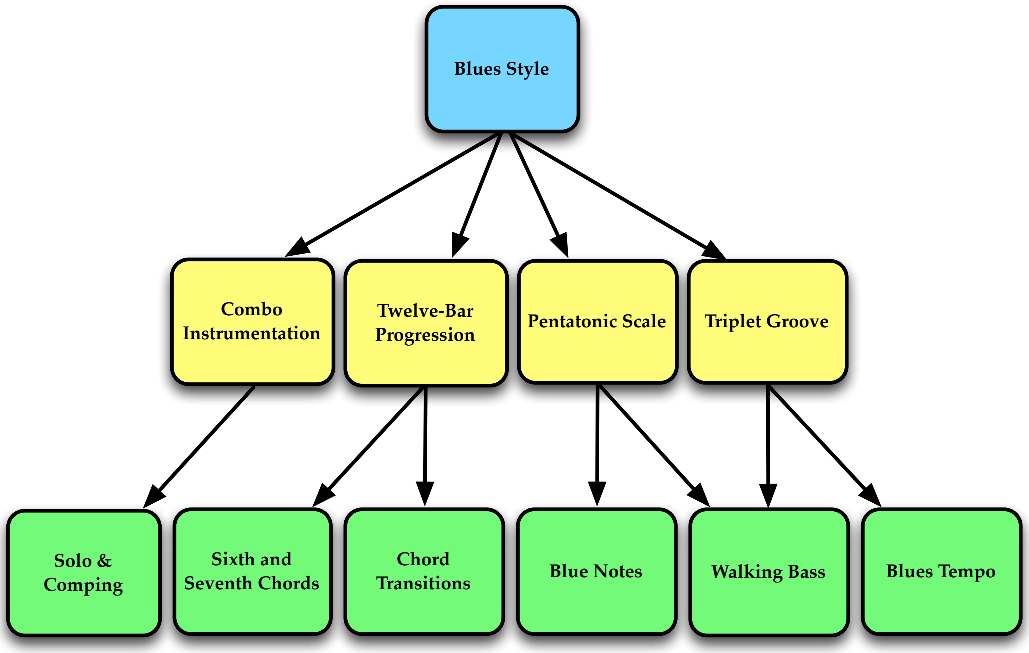


Figure 2.2: The *World Beat* pattern language that explains Blues concepts for the equally-named interactive music exhibit (adapted to Borchers [2001], p. 80).

Further Pattern Language Examples

PL on Web usability

The "wu" pattern language formulated by Graham [2003] organizes 79 patterns on web usability and website design. The aim is to collect knowledge from specialists in interface and interaction design, security as well as and technical stability and make it available for developers and development managers. The focused user groups often need to work in a multidisciplinary way and need to align the used technologies. Their challenge lies in meeting the requirements within the given development time and to ensure quality. In order to facilitate readability and navigation, the pattern are additionally divided into four sections of patterns treating

- preconditions to get started with a site,
- the enhancement of usability,
- details concerning technical aspects, such length of texts, existing standards, download times or content arrangement,
- workflows and security.

With a current problem in mind, the reader starts at an appropriate section and follows the links to associated patterns. The links represent the order in which the reader consults the individual patterns during the design process.

Start reading at current problem

Graham [2003] classifies the patterns in *abstract*, *concrete* and *abstract and concrete* patterns that are found everywhere in the pattern language. Thus, it does not present a hierarchy from abstract guidelines to concrete recommendations. Reader needs consult the pattern language during the entire development time. This way, the organization of the pattern language resembles a stepwise approach that continuously reveals details over time. This reflects the design and implementation process of a web site. The first two sections of the pattern language help fulfilling preconditions and cover basic rules for appealing and usable sites. Patterns in later sections deal with technical details for increasing the usability and technical stability of the site.

Consider patterns when necessary

In the domain of agile software development, Coplien and Harrison [2005] present four pattern languages giving advice on organizational aspects in order to be able to apply agile methods. They emphasize the need for patience and shape the term of "piecemeal growth" within an organization since changes in processes need time for acceptance. The authors encourage the reader to adapt and update the patterns to their current contexts and to mix the different pattern languages. They focus on the aspects of *organization design* which describes the settings necessary as prerequisite for agile processes and *organization construction* that gives advice on how to successfully implement the processes. The other pattern languages describe the aspects of *project management in conjunction with incremental growth* of the organizational structure and *organizational style concerning people and codes of behavior*.

Organizational patterns in agile software development

The four pattern languages are described as pattern layers that are supposed to help navigating the pattern language. Though, the layers are not separating the languages from each other. It is up to reader to make his own connections and links across the them. The single patterns build on each other as far as they are linked. At some places, patterns from the related pattern languages are introduced for explanatory reasons. The insertions are marked in the pattern language illustrations as graph structures. Readers have to look up the "external" pattern within the appropriate sub-language so that the contents need to be studied more thoroughly.

Separation into sub-languages

HCHI patterns

Schümmer and Lukosch [2007] follow the basic structure of Alexandrian design patterns by using natural language in order to describe solutions to specific design problems. Within their pattern language approach on computer-mediated interaction , they introduce a *scenario* field that sets the pattern's problem into an illustrating example context in order to increase understandability. *Checks* ask questions that try to help the reader to figure out whether the pattern representing a template solution is suited to the current design problem and *danger spots* show potential new problems that may occur when applying the pattern. They can be regarded as warning features trying to avoid the blind application of a pattern.

Evidence and interconnection

The closing sections of a pattern are named *known uses* representing the second part of a pattern's "proof" by presenting approaches in which the pattern is successfully applied. *Related patterns* link to relevant alternatives, patterns that are important for other stakeholders or patterns that describe further details.

Pattern Language Structure

Thus, Schümmer and Lukosch apply the DAG structure rules as defined by Borchers [2001] in which nodes may have an arbitrary number of incoming and outgoing connections to other nodes, i.e. patterns. At a high level of detail, there are no more related patterns and therefore only incoming edges. In order to increase readability, they structure the pattern language in two dimensions as shown in Figure 2.3.

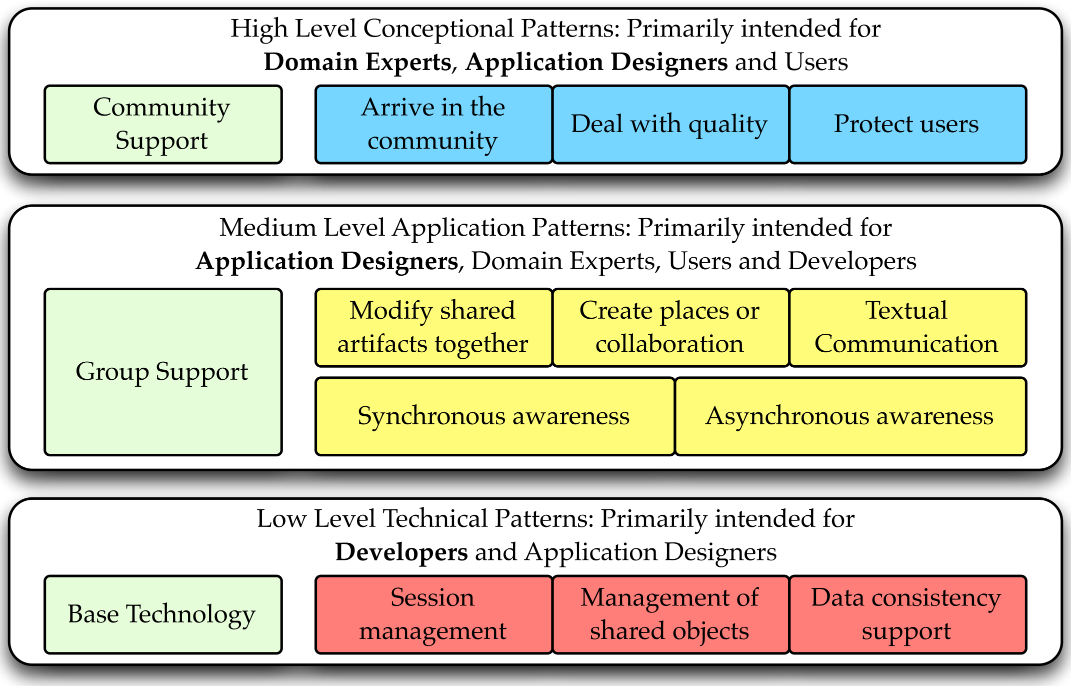


Figure 2.3: The clusters topology of the pattern language presented by Schümmer and Lukosch. Illustration adapted from Schümmer and Lukosch [2007], page 43.

On the *vertical* dimension, they introduce three layers of technical detail described by a pattern as follows:

Technical detail

- *High level conceptual patterns* describe what a computer-mediated collaborative interaction should look like. This description is understandable for all stakeholders, starting from the targeted user group, passing application designers and reaching to software developers.
- *Medium level application patterns* incorporate technical solutions and mention possible technologies or processes to be used. At this level, some details may not yet be clear to the user group or the application designer. However, this does not hinder the understanding of the whole concept of the intended application. The technical character of the patterns is more relevant for the technical implementation.
- *Low level technical patterns* primarily assist software developers in realizing specific solutions. For this kind of patterns, there are no more usage-related conceptual suggestions given. Usually, other stakeholders are not interested in the technical details that run in the background of the system.

On the *horizontal* dimension, the patterns are clustered as conceptual groups covering different development aspects, like "dealing with quality", "protecting the user" or "creating places for collaboration". With the help of this structure, readers receive assistance in finding their own knowledge and interest level within the considerably large language consisting of 72 patterns. After the initial orientation, readers can concentrate on a more detailed level of topics they are interested in by browsing the language via the horizontal classification.

Conceptual groups

van Duyne et al. [2007] organize their pattern language that consists of 107 patterns about successful website design in groups according to different aspects of website design. Starting with suggestions for different site genres and navigation, they describe details about content management, trust, credibility, e-commerce as well as advanced topics dealing with site layout up to page optimization for mobile devices. The authors encourage to consult the pattern language during an iterative design process. They recommend special patterns for the phases *design*, *implement* and *analyze*. The authors intensively make use of links between patterns in different groups. Readers have to decide whether the links are relevant for their current design situation.

PL on website design clustered in groups

Kunert [2009] presents a pattern language for designing interactive digital television applications . The patterns are grouped one-dimensionally on different aspects of digital television application starting with patterns for laying out screen pages, creating concepts for navigating the application, assigning remote control keys and then discussing basic functionality like starting, exiting and hiding the application. Other patterns deal with the presentation of content, user participation, text input, help screens, sections, accessibility, personalization as well as specific user groups. For the latter aspect, children as a user group is examined more closely. Kunert misses to introduce

PL for interactive TV applications

an illustrating navigation structure but lists the aspects around which the patterns are arranged and asks the reader to follow the links to related patterns.

PL for designing interfaces

Tidwell [2011] presents comprehensive patterns for interface design . The work presents concepts that were gathered since over ten years originating in the online available pattern language "Common Ground"[3] whose site was last updated in 1999. The described concepts that deal with successful human-computer-interaction design were reformulated and extended in her book "Designing Interfaces" in a current edition from 2011. The patterns are grouped by different aspects that deal with the organization of content, navigational elements, layout of elements, lists, actions, visualization of complex data, dealing with user input, social media, mobile interfaces as well as visual styles and aesthetics. One special pattern group that should be read first and be kept in mind when examining the advice from the other patterns is about user perception and user behavior. Tidwell points out that the described concepts are the foundations for the presented patterns and are relevant for interface and application design. Although patterns are partitioned into conceptual groups and related to each other, the author does not provide a visual organization structure of the whole pattern language.

Pattern Language Examples in Non-Technical Domains

Patterns in education

Also in non-technical domains, pattern languages are applied successfully. So, e.g., in the educational domain as presented by Bergin et al. [2012] who creates a pattern hierarchy along the time dimension. The derived pattern language follows the original Alexandrian structure. After providing introductory patterns that form the basis of the explained concepts, advice for structuring educational contents over weeks, hours and minutes is given. The patterns were distilled based on the *Pedagogical Patterns Project* that supports teachers in questions about group-based learning with collaboration technology (cf. Bergin).

Patterns for organizational structures

Manns and Rising [2005] present ways to introduce new ideas into an existing ecosystem with its existing organizational structures . Their pattern language *Fearless Change* relies on social structures and requires practices that establish trust in new goals. The patterns are interconnected when explaining the context in which the individual pattern should be applied. Thus, the patterns that could have been applied before the current one or that need to be understood are referred to. While discussing the proposed solution and resulting situation after applying a pattern, references on alternatives and additional patterns are given.

Pattern organized around phases

The pattern language itself is presented in a plain way, i.e., an alphabetically ordered list, without any structural illustration. However, in the preceding chapters, the authors guide the reader through the pattern language by grouping patterns by steps to be taken over time. This way, the pattern list serves as a compendium for look-up purposes. Readers dealing with the handled topic of the language are told which steps

[3]http://www.mit.edu/~jtidwell/common_ground.html

in general need to be taken in order to make changes within an organization. First, initial steps are described followed by recommendation on how to organize meetings and convincing colleagues as well as superiors. Later, advice on dealing with resistance from the people within an organization is given. For every phase, a special collection of patterns is recommended to be read first. From the pattern collections, readers may follow the references or stick to the presented ones in a first iteration over the language.

Patterns on reaching professionalism

Shimomukai et al. [2014] organize 31 patterns on taking action in the field of social entrepreneurship . The pattern language describes different phases that concentrate on first creating a mindset based on a micro vision. Later patterns deal with scaling out the actions to take in order to reach professionalism. The lower level phases deal with self-empowerment in order to get the change started. The later phases deal with the change making process itself. The authors illustrate their pattern language along a tree shape. Thus, the patterns are supposed to be read bottom-up, starting at the two phases in the basic level and ending at the treetop with the gateway to professionally applying the formulated guidelines for successful social entrepreneurship. The patterns themselves are formulated in a very condensed way such that the reader can skim through the contents quite easily. However, the discussion of solutions concerning problems and forces is treated very tersely.

2.2.3 Pattern Structures

Commonalities

After the presented survey of different pattern approaches in different domains, this section describes the pattern structure that commonly applied. Depending on the domain in which patterns are formulated, additional fields are introduced. Especially in interface design and architecture, examples and illustrations support the understanding of given solutions. For the solution illustration, most authors prefer abstract or hand-drawn pictures in order to avoid real examples and to retain the pattern's abstractness and applicability for similar problems.

Reduced canonical form

In adaption to Gamma et al. [1994] and Buschmann et al. [1996], the *Reduced Canonical Form* is the minimalistic format for a pattern. This common root of all pattern structures concentrates on:

- a *Name* that clearly expresses the central idea of the pattern.
- a brief description of the *Problem*.
- a *Solution* to the problem.

Varying description lengths

Several pattern authors extend this format in different ways and lengths of description. Alexander [1977] and Borchers [2001], for example, provide very detailed discussions of the problems and solutions. They explain sources of the problems and give reasons why the proposed solution works. Iba and Isaku [2014] however, propose a very condensed form of formulation. They prefer short and distinct descriptions of the different fields in order to make it easy for readers to consume many proposed patterns in shorter time. In their view, patterns are a medium for exchange an communication about experience with patterns also beyond the given formulations.

Common pattern fields

From the analysis of pattern language approaches, mainly the following fields are used in order to structure the formulated design patterns. The denomination of the fields slightly changes in the individual cases:

- A *Name* that clearly expresses the central idea of the pattern in order to serve as vocabulary item. This way, it becomes easier to remember the pattern and recover its central idea, especially in conjunction with an illustration (see below).
- The *Ranking* reflects the author's subjective view on the invariance of the pattern. It supports the reader to decide whether the pattern is common to be taken or it is recommended to consider alternative approaches.
- The *Context* describes in which design situation the problem occurs ranging from high-level concepts up to details that occur late in the design or conceptual process.
- *Illustrations* are closely connected to the name, such that the reader receives a sensitizing example for the pattern's application and can quickly grasp the main idea of the pattern.
- *Problem and Forces:* This part discusses the problems occurring in certain situations and what kind of influences (forces) are to be taken into consideration. Borchers [2001] adopts the notion of forces from Alexander [1977] and Murchison [1930] where forces are bound to physiological, social and economical forces as well as cognitive psychology. Thus, forces are influences that encourage users to do or refrain from certain actions. These can be conscious or unconscious ones. For example, it is a natural force that people tend to walk into light areas of a room or to gather at a sitting location like a table. Thus, there may be forces "pulling" people into different directions in a worst case.
- *Examples and Discussion* intend to help the reader to inductively follow the solution that is suggested by the pattern suggests.
- The *Solution:* is the general message of a pattern generalizing from the concrete examples to an abstract solution to be applied in different contexts.
- A *Diagram* summarizes the described solution in a schematic and condensed way in order to make it easy to remember and to cover the key elements of a pattern again. It may range from sketches over photographies to formal diagrams.
- *References* connect patterns to each other in case they are related by presenting additional information or alternatives to the proposed solution.

2.3 Pattern Mining and Assessment

Qualities of a pattern

Collections and pattern languages represent a solid source of knowledge for experts to use for teaching concepts within the domain, novices and non-experts who want to learn about it. As communication medium, they can be used to act as common vocabulary within expert groups but also, in case they are formulated without domain-specific jargon, as cross-disciplinary language. Successful collections are long-living and known by many and eventually updated by new finding depending on the authors attitude and interests.

Different approaches for pattern derivation

This section discusses the question of the origin of design patterns. Author groups follow different approaches to identify, formulate and validate patterns and arrange them in a pattern language . Personal exchange and focused group discussion with other pattern and domain experts is followed as well as approaches that bring together many contributors in large-scale, personal workshops or interconnect a pattern community via online available portals. Other approaches integrate the derivation and application into the whole development process in iterative manners.

2.3.1 Exchange Within Communities

Specialized pattern conferences

A very strong and active pattern community that establishes a forum for pattern authors to evaluate their pattern assemblies is the Hillside Group[4] that was founded in during the OOPSLA 1991 conference by Ken Auer, Kent Beck, Grady Booch, Jim Coplien, Ward Cunningham, Hal Hildebrand and Ralph Johnson[5]. This group regularly organizes *Pattern Languages of Programs* (PLoP) conferences . These events are taking place worldwide including North America (*PLoP* or *ChiliPLoP*), South and Latin America (*SugarLoafPLoP*), Europe (*EuroPLoP* and *VikingPLoP*), Asia (*AsiaPLoP*). The first PLoP conference for India (*GuruPLoP*) is announced for 2013.

Detailed pattern discussions in workshops

The key tool of evaluating pattern formulations is represented by *Writers' Workshops* that are regularly arranged within these communities in which new patterns are formulated, read out to the audience and then discussed by all. This process is intended to help the writer to optimize the quality of a pattern and to check its validity. The whole workshop is organized in a friendly setting and each author's work is always honored. It is important that potentials for improving the submission is outlined. Criticism must take place in a constructive manner.Web-based community portals allow remote discussions in a distributed manner. This way, there is both - a personal and remote character supporting the continuous discussion in the field of design patterns.

[4]http://hillside.net

[5]http://oopsla.org; OOPSLA, the conference on Object-Oriented Programming, Systems, Languages, and Applications takes place under the umbrella of the SPLASH conference since 2010. http://splashcon.org; The conference on Systems, Programming, Languages and Applications: Software for Humanity.

Extraction from usability work

Similarly, the pattern language published by Graham [2003] on web usability was created. Every pattern that was influenced by others during workshops and interviews has an acknowledgment section naming the involved persons. Thus, the patterns are created based on distilled pieces of knowledge from different work in usability, cognitive psychology, user interface and interaction design that is described in books and guidelines. After the pattern extraction process, they were discussed with others that contributed new aspects, eventually leading to new pattern ideas.

Gathered from professional experience

Manns and Rising [2005] also state in the preface of their pattern language that the collected knowledge was gathered over years of professional experience and exchange with colleagues and other professionals in the field who made similar experiences or applied proposed patterns. The feedback and discussion took place during workshops and was focused during PLoP conferences.

Based on user feedback

Similarly, van Duyne et al. [2007] present their pattern language as a large collection to consult during work. They clearly state that their publication does not claim for completeness and correctness but reflects their experience from practical projects as well as academic backgrounds. The authors are aware that continuous integration of user feedback is necessary to keep pace with the rapid developments of web technologies.

Central pattern repository

A very ambitious approach is envisioned by Gaffar et al. [2003]. The present *Montreal Online Usability Patterns Digital Library* aims at integrating many different pattern languages and pattern formats into one central repository. According to the defined *Seven C's Method* patterns should be integrated by applying the following steps:

- *Collect* patterns from different research work and gather them centrally.
- *Cleanup* the collection by identifying contradictions between proposed solutions. Merged similarities or find alternative solutions.
- *Certify* the patterns by assigning concrete domains in which they can be applied.
- *Contribute* new emerging patterns from other work.
- *Categorize* the patterns within the collection.
- *Connect* patterns to each other by semantic associations, ideally by using an ontology.
- *Control* the pattern repository with machine-based tool support. This affords the patterns to be formulated in an additional machine-readable format.

Automated pattern management

In later work, Gaffar et al. [2005] propose a more abstract way to formulate user-centered tasks concerning pattern formulation, management and interaction with pattern language in terms of searching, browsing and combining patterns. The overall aim is to partially automate pattern-related tasks such as validation, comparison, discovery and combination. A first step is the application of an XML schema that expresses the conceptional division of *intrinsic* information, i.e. the considerations on problems and solution that are described by a pattern itself. *Extrinsic* information refers to the relation among patterns that may express similarity, competition or mutual contradiction. Information on the *assimilation* of patterns focuses on where, when and how a pattern can be applied during an integrated design process.

Online Discussions

Web-based discussion on initial platform

Online communities make use of web-based platforms for exchanging and discussing experience. New members in the role of readers, contributors and validators are welcome. Especially in the domain of interface design, online collections of design patterns have emerged. For these kinds of collections, an initial pattern collection is published online and opened for discussion. Thus, feedback from the community about pattern readability, experience from the application as well as supporting examples and implementations of the pattern are externally contributed. This way, the author group is still responsible to set up the initial structure but is able to react on feedback quicker and make this feedback available for the whole interested community. The forum-like approaches potentially attract many interested persons and contributors but the site administrators need to take care about the quality and appropriateness of the contributions. Experts on pattern writing are intended to cooperate with contributions coming from domain experts.

Examples mostly support UI design

Most online examples can be found for commonly applicable user interface (UI) problems. Thus, Martijn van Welie's library containing *Patterns in Interaction Design*, the *UI-Patterns* collection and the *Pattern Tap* site deal with collections about single aspects of interface design (cf. PatternTap LLC, Toxboe, van Welie, Yahoo! Inc.). The navigation structure is mostly based on a categorized linked list and has the character of browsing wikis with the help of breadcrumbs. Readers mostly have to remember where inside the pattern language or collections, respectively, they are. Still, it is possible to navigate towards specific design problems in a straight forward manner. The aim of these platforms is to give support on specific problems and less on providing the large situational picture of the domain as shown in the pattern language examples in Section 2.2. The *Android Patterns* (cf. UNITiD Interaction Designers) propose solutions for design problems on mobile devices. The navigation starts with a general topic, e.g., "dealing with data" and then offers the reader a set of related questions for that context. Breadcrumbs show the current context and taken path within the pattern language.

Public and private pattern libraries

The online pattern libraries *Quince* and *Patternry* represent exceptions to the public community process. The free version shows public patterns but tries to convince the user to register for a chargeable extended license that provides collaboration spaces restricted to project teams (cf. Infragistics and Lammi et al.). This way, patterns and experience can be restricted from the public and only used internally. However, the collected data is still stored on foreign servers. The publicly available version of Patternry offers many up-to-date design suggestions and additional "useful links" to current libraries that implement the suggested solution. Since they mainly deal with solutions for websites, HTML and CSS code snippets are prepared to be copied and adapted to the reader's needs. A workflow for adding and refining patterns is not explicitly defined. Groups have to find ways to organize themselves.

Feedback channel but need of administration

The potential of online discussion approaches lies in the vividness of contributions and the availability of fast feedback channels that are visible for all participants. New ideas, experiences and insights can be communicated by every participant. Still, the risk of doubling patterns in the different sites exists. This can partially be handled by user-comments on the patterns and moderators that maintain the library.

Workflow for distributed design teams

Leacock et al. [2005] formulate and implement a pattern formulation workflow for the Yahoo! Pattern Library[6] with the aim of managing results and coordinate design efforts that take place in spatially distributed tams in one central repository. The workflow is aligned to the Yahoo! design process itself covering the aspects of identifying patterns and planning their formulation by setting a time frame and assigning authors. After a pattern is formulated, it gets published within the repository where comments on the formulation are collected and the author has the chance of incorporating them into the pattern formulation. As soon as the author is personally satisfied with the current formulation, he marks the patterns as "ready for review". After this point, a rating mechanism on the pattern is triggered that concentrates on the pattern's understandability, correctness, repeated application, supporting research and relevance to the repository. Ratings are provided by a review committee consisting of member from different business units and backgrounds. The author has the chance to react to the ratings and make changes to the formulations that are reassessed by the committee during the next review iteration. Submissions, comments updates and reviews are communicated in a notification roll up that summarizes the activities for authors and subscribers interested in patterns and change events. Since the process itself is embedded into the workflow of the Yahoo! design team, tasks for identifying, writing and reviewing patterns can be assigned. Considerations about concrete incentives are briefly described to strengthen the motivation to ensure the quality of the contributions. The patterns are organized by browsable categories. Users can navigate the pattern language by categories, links between related patterns or type fields of a pattern like "task", "application" or "device".

2.3.2 Active Pattern Mining

Pattern extraction in large workshops

Iba and Isaku [2014] present a holistic pattern mining process based on large groups of domain experts that consequently mind patterns based on their knowledge. They formulate the process itself as a pattern language that guides the reader stepwise through the different states of the process while discussing problems that may occur. The patterns express gathered experience from several workshops that were held at Keio Uni-

[6]http://developer.yahoo.com/ypatterns

versity, Japan[7]. The described process asks the participants to consider the domain in which pattern mining should take place as a whole and then assign experts on certain aspects of the domain. Each group is allowed to concentrate on its own experience but also contribute to thoughts presented by other groups. This way, the finding process shall be fertilized better. Examples shall be given from the first minute to support the formulated ideas. The process foresees the following steps:

- In the beginning, every idea for a pattern is written on a post-it note. Then, the notes are placed on a free space on the table - if possible, close to a note with similar content. After the initial drafts, ideas need to be described further, preferably in an imperative form.
- In a second round, the pattern ideas are grouped and clustered. This way, duplicates could be merged or filtered out. The clustering may require a complete rearrangement of the layout. Participants are asked to spatially arrange the patterns such the similarity and close relations are expressed by proximity.
- In a third step, islands of very close concepts are merged and labels are given. Connections between the clusters are discussed and drawn on agreement. This step reveals an initial pattern language structure, pattern ideas and the interrelation among each other.

Expenditure of time

The proposed method is very consequent to exchange and formulate ideas and knowledge of the expert team. Though, the time needed for this process and the need for avoiding interruptions may represent a high obstacle for the application in environments with high pressure of productivity and strong workload.

Evaluation

Kunert [2009] applies a smaller scale approach by first identifying interaction problems based on analysis of usage contexts users' tasks and requirements, existing guidance for certain platforms or domains and designers' requirements for guidance. Then, solution concepts are created based on existing knowledge, i.e., guidelines, experience or existing patterns, and new ideas. After building prototypes and extensively evaluating them, pattern solutions are approved. Within the pattern language, support for patterns is given by examples from literature or existing patterns and the results of the custom prototype evaluations.

Collaboration and exchange

Even more focused are the approaches for deriving design patterns as described by Borchers [2001], Coplien and Harrison [2005], Graham [2003], Scott and Neil [2009], van Duyne et al. [2007] and Tidwell [2011]. They state that the patterns were assembled over long experience times together with research colleagues. During presenting formulated patterns, feedback from focus groups was collected, e.g., during writers' workshops (cf. Section 2.3.1), and integrated into the pattern language. Remarks on the patterns from colleagues working in the field as well as mailed feedback on the publications were taken into account for new editions of the pattern languages.

[7]http://www.keio.ac.jp

2.3.3 Mining During Engineering

Application and improvement of existing patterns

Within the scope of systems engineering for safety-critical environments, the Frequentis process as described by Grill and Blauhut [2008] starts with the creation of an early draft of the user interface. This step is based on previous workshops that were held to analyze the requirements with the users. During the latest design phase, the design is evaluated with the users based on their expectations. This way, the design of the user interfaces are continuously improved. After settling the UI design with users, design experts have the possibility to access a pattern library, containing important patterns identified from previous projects. Thus, existing patterns are reused and further refined beyond the current project scope. New findings are inserted as patterns and validated over time. However, the patterns are only applied once during the final interaction design phase of the current project. Validation is based on the number of repeated applications of a pattern. This way, the patterns are categorized into *regularly used*, *proven to describe a good interaction design approach* and *relevant* patterns. Mostly, aspects concerning user interface designs are captured by the patterns. Other project aspects and reasons for the pattern formulation are not given in a larger scope. They are only mentioned within the pattern description. Contributions from outside are not foreseen. Only project internals have access to the accumulated knowledge which is understandable in an industrial context but prohibits public discussions.

Pattern generation during development

de Rore [2009] makes use of software design patterns as a means to increase software productivity and quality. During the design of her studies and performing measurements on software productivity, she describes that patterns need to be introduced and pattern writing needs to be taught. For pattern introduction, she follows the *Fearless Change* approach as described by Manns and Rising [2005]. This way, the development community learns about the pattern topic itself and the application of patterns. While discussing the patterns, learners and teachers refine and eventually update the pattern. In the scope of her work, de Rore also introduces a first attempt to generate patterns during the engineering process by documenting found solutions in the pattern format already. Thus, she makes a distinction between early formulations that are collected from the beginning of the project and evaluated solutions that define patterns over time that are distilled after a longer period of time. One finding is that the parallel documentation needs time, but participants are motivated more easily since they do not have to invest extra efforts near the end of the project. De Rore introduces the idea of documenting patterns on the architectural level. There, architectural decisions are documents in the pattern format. However, this process is not fully evaluated but introduced as an alternative idea to pattern mining.

From observations to best practices

Averbakh et al. [2011] address the problem of knowledge documentation and experience sharing in globally distributed software projects. Their approach is implemented and evaluated in the scope of the GloSE project[8]. Averbakh et al. attempt to integrate the formulation of knowledge and experiences during the engineering process work.

[8]http://www.se.uni-hannover.de/pages/en:projekte_glose

Every project member is allowed to formulate findings and working solutions as observation sheets that are assembled in a wiki. Over time, the collected observation sheets are refined by dedicated project members, called *experience engineers*. They extract often repeated passages from the observation sheets and merge them into consolidated documents. In follow-up steps, the documents are rephrased more formally and categorized. Afterwards, recommendations are formulated that can be approved by the original authors as published best practices and submit these ideas to a project-wide knowledge pool of best practices. However, the evaluated finding are not organized within a graph structure. The generated documents reside within a collection without relations to each other or hierarchies concerning abstractions or concretizations.

Role model for content protection

The approach also takes into account the possibility that partners refuse to share their experience due mistrust within the consortium or the fear to loose an strategical advantage if sharing their knowledge. For that reason, a role model based on Schneider [2000] is introduced such that only approved best practices are available beyond the individual organization's scope. During the process of extracting recurring knowledge and merging them to recommendations, the contributions are anonymized and treated confidentially. These extra steps put a higher level of complexity to the whole process with need for special roles and protection mechanisms. This also influences the flexibility and vividness of knowledge extraction, formulation and sharing. Although the authors do not explicitly apply the notion of design patterns, the approach is considered as relevant for knowledge extraction and documentation.

2.4 Conclusion

Patterns used to communicate knowledge

This chapter introduced the concept of design patterns which originates from architecture. The key qualities of patterns are to convey design knowledge also for non-experts within a domain by formulating them in an informal way in prose language. A pattern represents knowledge gathered within a specific domain, discusses the context in which it could be applied and explains ways to solve a current problem. The solution described by a pattern is kept abstract enough to be reused in similar problem situations. The target group may vary from domain experts over students to interested parties without any specialized domain knowledge. Alexander [1977], Borchers [2001] as well as Schümmer and Lukosch [2007] demand that pattern languages must be understandable for non-expert stakeholders within a project. This way, patterns act as a communication medium as well as a shared vocabulary between different layers of professionalism. The defined pattern fields help readers to mentally structure the content and directly skip to parts of interest.

Comparison with formal methods

From a brief overview of alternative ways for formulating patterns by means of mathematical logic or ontologies, the need for informal formulations in the interest for the project-wide communication was strengthened. Therefore, the approach presented in this thesis makes use of informal pattern formulations.

Patterns in many domains

In many different fields where knowledge and experience can be captured, patterns have proven to be a useful approach. Gamma et al. [1994] first applied the concept to software engineering. The solutions described in this work are reused and taught in and software engineering. Borchers [2001] and Tidwell [2011] transferred the concept to interaction and website design. Other related disciplines followed their approaches as shown in the presented survey. Apart from architecture and computer science, pattern collections for organizational learning or education were presented as prominent examples. With regard to the focused domain of emergency response, existing pattern approaches and related work that may lead to patterns were discussed in addition.

Pattern collections and languages

Pattern collections cluster and structure derived sets of patterns, e.g., by purpose, context or behavior, making it easier for readers to mentally order the patterns and focus on current needs. Pattern languages interconnect single patterns and therewith provide a context, in which a single pattern can be applied. References to related patterns reveal additional details as pattern combinations or alternatives. Thus, pattern languages can be represented as directed acyclic graph structures according to the formal definition for pattern languages given by Borchers [2001].

Pattern mining approaches

This chapter also discussed ways to discover patterns by applying different mining methods based on writers' workshops within small groups during pattern-related conferences. Alternatively, specialized pattern formulation workshops can be organized with domain experts. Approaches that involve large groups of pattern authors last for several days, as described. Discussion forums in online-available pattern libraries allow for community-based contributions of pattern knowledge and feedback on existing contributions. Other approaches within project work groups are based on documenting best practices during the development process.

Problem scope and testbed

The next chapter elaborates the problem scope of distributed joint research projects that is mainly tackled by this thesis and presents the characteristics of this kind of project structures. Identified challenges in general regarding knowledge transfer are presented together with the description of the peculiarities of the concrete project BRIDGE that is used as testbed for the developed approach.

Chapter 3

Engineering in Distributed Joint Research Projects

Efforts and overheads

The research efforts of this thesis concentrate on large-scale, distributed joint research projects in which a large number of partners and participants needs to be managed and a high degree of currentness of information has to be ensured in order to collaboratively achieve the project's aims. Additionally, the communication overhead concerning strategy, requirements or concepts changes must not be neglected. Large amounts of documentation must be created in manageable amounts and sizes in order to be effective. In current situations but also in retrospective, reasons for design decisions must be understandable and reproducible.

Research projects structure

First, this chapter describes the structural composition of research projects together with the problems concerning the integration of existing knowledge, its maintenance throughout development as well as its exploitation.

Challenges for exchange

Second, challenges for knowledge exchange between the project members are discussed. Different engineering methods among the partners lead to additional inconsistencies in documentation formats and activities. The big communication overhead results in the danger of unbalanced information states between the project partners. As a consequence, important facts do not find their ways into the designs and concepts.

Concrete project as testbed

Third, this chapter presents the project BRIDGE as a concrete example for a distributed joint research project situated in the domain of emergency response (ER). At the same time, it serves as testbed for the creation, refinement and evaluation of the concepts presented in this thesis. BRIDGE strongly applies user-centered engineering techniques together with the creation of concepts and technical realizations combined with continuous validations. In the project context, a survey on pattern approaches for the ER domain is presented.

3.1 Project Structures and Development Efforts

High efforts for project management

When starting a new large scale project the recommended management process according to acknowledged strategies like the "Project Management Body of Knowledge" is to define the project scope and a management plan by taking into account the project's requirements and possible risks (cf. Institute [2008]). Then, the implementation and change request process starts which ideally does not follow the traditional waterfall model (cf. Royce [1987]) with its fixed and non-reversible phases but a more iteratively-driven process, that should at least be based on the spiral model introduced by Boehm [1986]. However, within the project contexts in the focus of this work, there is the need of putting lots of management, coordination and communication efforts into the project. The distribution of work as well as the high degree of the participants' independence are further aspects that make the enforcement of rules and practices harder to introduce. The consortium continuously needs agreements and negotiations throughout the project runtime.

Practical experience

The ideas and concepts this thesis discusses are based on the author's practical experience within distributed joint research projects gathered over a period of six years at the Fraunhofer Institute for Applied Information Technology FIT in Sankt Augustin, Germany[1], as shown in Table 3.1. Throughout the projects, the author applied user-centered engineering methods as proposed by Nielsen [1993] and Greenbaum and Kyng [1991] that were combined with domain analysis and requirements elicitation. Domain analysis took place in an iterative manner by making use of participatory and ethnographic design approaches (cf. Suchman [2007] and Hughes et al. [1992]). Prototypes were continuously conceived, evaluated and adapted to reach accepted solutions. Exploitations as publications, project reports and presentations were extracted and given by the author in parallel to the developments. All efforts were based on the aim of understanding the user group and to familiarize with its tasks, needs and working environments as described by Shneiderman and Plaisant [2004].

Research projects characteristics

Research projects often have a highly distributed character in which different institutions, small-medium-sized and large enterprises collaborate. Each partner influences the project with his expert knowledge and individually approved methods for engineering and validation. Table 3.1 gives an overview of the project experience gathered over the last six years. The duration of the different research projects lay between three and four years, depending on their type. Networks of Excellence (NoE) and Integrated Projects (IP) usually have a four-years runtime whereas Collaborative Projects (CP) run for three years. The European Commissions (EC) favors cooperation, knowledge exchange and taking the advantage of synergy effects between different expert groups within the research consortium[2]. The four nationally funded projects (NF) AILB II, III, Gateway and DeafTrain each a had runtime of three years. Since every partner brings

[1] http://fit.fraunhofer.de
[2] http://ec.europa.eu/research/fp7/index_en.cfm?pg=cooperation

Project	**Type**	**Topics**	**Experience**
InterMedia	NoE	Multimedia content handling, interaction and transformation.	2007 - 2010
LinkSmart	IP	Middleware development for embedded systems and sensor integration.	2007 - 2010
AILB, Gateway, DeafTrain	NF	Learning and information portals for preparing hearing impaired students for the labor market and university studies.	since 2008
BRIDGE	IP	Emergency response and collaboration support, system and HCI design.	since 2011

Table 3.1: Overview of the author's participation in research projects since 2008.

in his specific competences and working processes, project management must be very flexible but still follow a budget plan and schedule.

Division into work packages

Within a joint research project, the number of participants quickly spans up to twenty partners each introducing two to four project associates[3]. Accordingly, the amount of consortium members that need to be managed and coordinated quickly reaches 50 to 100 people. In order to coordinate the efforts between the different stakeholders of the projects, up to 15 different work packages (WP) are specified that focus on specific project tasks. One specific work package is responsible for managing the entire project and reporting. The other work packages focus on domain analysis, eliciting requirements, implementation, application and interface design, specifying software and system architectures, performing technical integration, testing and validating as well as exploiting the project results. Figure 3.1 shows a conceptual overview of a possible work package distribution. The individual results and therewith the interdependence between the work packages are shown as directed arrows. Mutual influences may exist as shown in the case of *WP: Req. - Engineering* and *WP: Domain Analysis*. *WP: Management* is shown as embracing structure as it is interconnected with all available work packages. The work packages that are relevant for the discussion are interconnected by solid strokes and reflect the iterative design approach according to Nielsen [1993].

Knowledge in project phases

Figure 3.2 shows a high-level process of iterative engineering tasks that are related to the knowledge handling process within a research group. When starting a project, knowledge is *generated* in the requirements engineering phase that is influenced by domain analysis and the reuse of existing findings and experiences. During the development and evaluation phase, knowledge is *refined and extended* by iteratively validating created concepts and prototypes. At the end of the project, knowledge is *exploited* for productive implementations that are tailored to specific solutions and for future research projects. The three different phases are discussed in the following.

[3]The CORDIS database provides an overview of the currently running research projects in the EU: http://cordis.europa.eu/projects/home_en.html

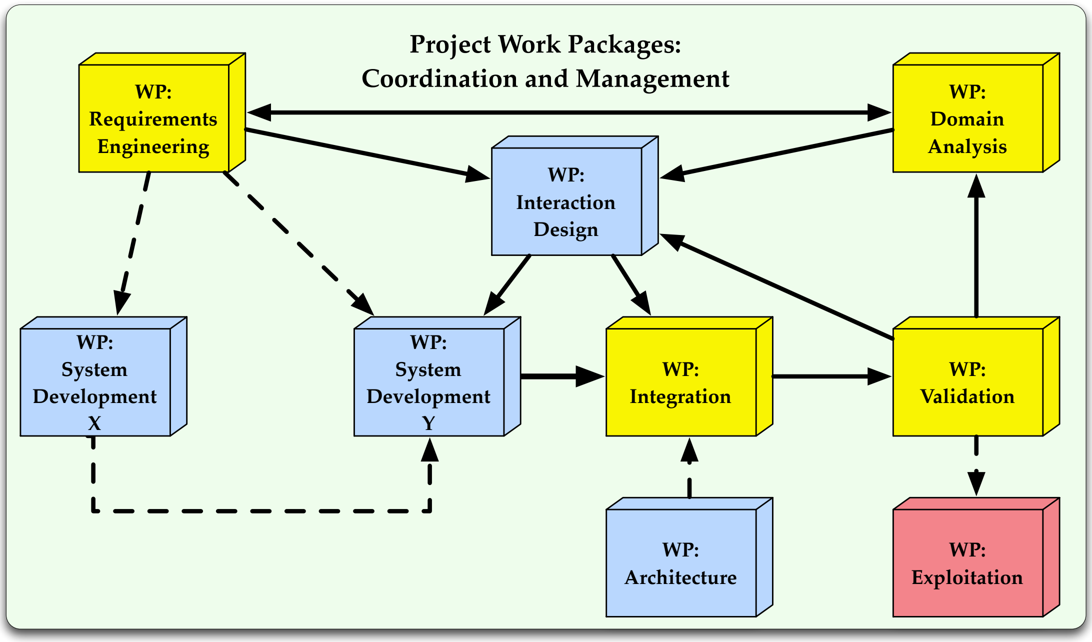


Figure 3.1: An example distributed research project structure and important influences and interdependence between the different work packages.

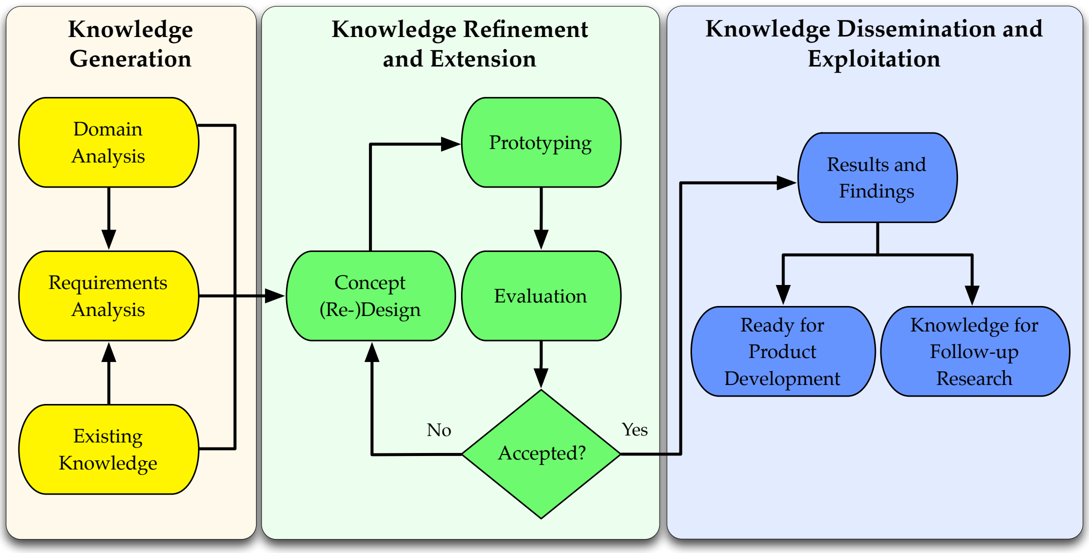


Figure 3.2: A conceptional illustration of an iterative design process within a research project.

3.1.1 Knowledge Generation

Requirements analysis is the central element of knowledge generation at project start. Though there are different kinds of requirements that need to be revealed with different techniques as described in the following. Domain analysis and existing knowledge from the project partners and stakeholders further influence the elicitation of functional and non-functional requirements. The latter represent more peculiarities of the domain and the target user group and are very important for application and interaction design.

Requirements contain knowledge

According to Kano et al. [1984] there are *mandatory requirements* that absolutely need to be met by the application so that it is accepted by users. Additional *attractive requirements* and features the product offers support the advantage of the product in contrast to other existing approaches and can make it more successful. In between, there are *specified requirements* that the customer explicitly demands for the product and that can be specified, measured and technically formulated. Figure 3.3 illustrates the correlations between the different kinds of requirements, as further explained in the following.

Types of requirements

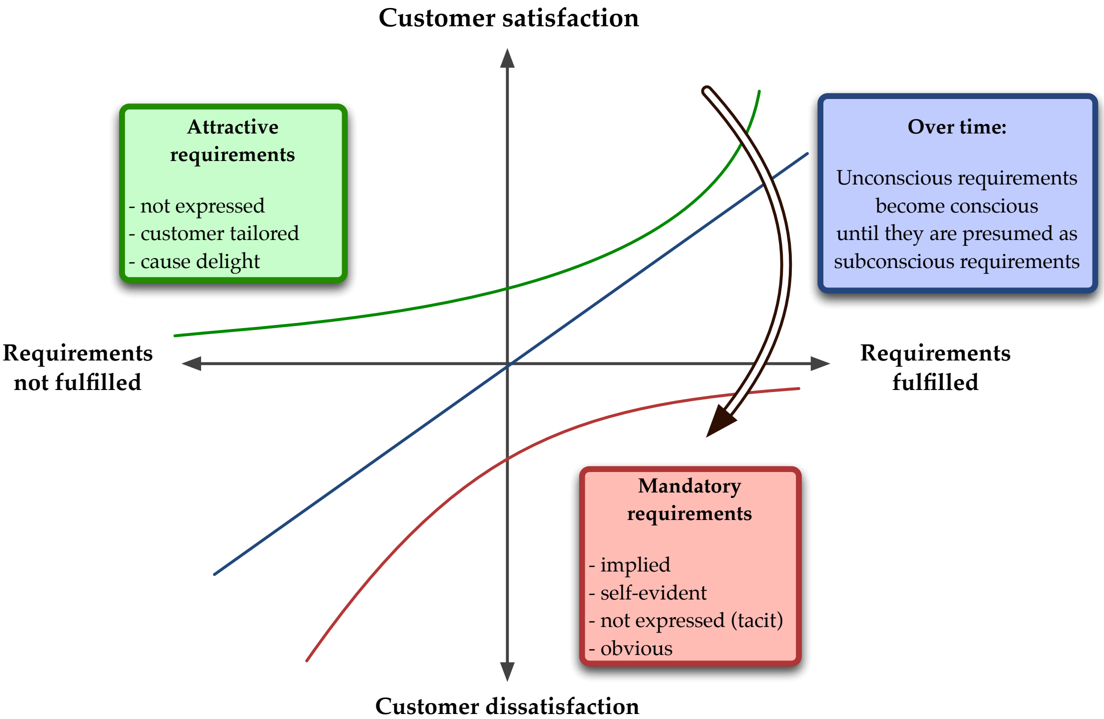


Figure 3.3: The Kano diagram illustrates the correlation between mandatory and attractive requirements (adapted to Kano et al. [1984]).

Mandatory requirements are often not directly expressed since they describe habits the users are no longer actively aware of (cf. Goguen and Linde [1993]). Observatory methods help to discover this *tacit knowledge*. Domain representatives are watched while performing their work. Participatory design approaches combine observations with

Tacit knowledge

ethnographic methods that try to involve domain and system engineers into the task performed in the considered domain as described by Suchman [2007] and Hughes et al. [1992]. These approaches represent promising ways for the domain engineer to learn about tacit knowledge (cf. Sommerville et al. [1993]). Kensing and Blomberg [1998] discuss the integration of a participatory design approach that involves the user at a very early stage in the design process in addition to Nielsen's approach of iteratively designing prototypes in which the user is not only asked for evaluating the design outcome but also takes an active part in the requirements engineering process and system behavior specification.

Creativity methods

According to Maiden and Gizikis [2001], creativity techniques have proven as successful in order to learn about attractive requirements. Very common methods are *brainstorming* and the *6-3-5-method* in which six participants each create three ideas that are circulated and refined five times (cf. Rohrbach [1969]). De Bono [2006] introduces the *six-hats-method* that allows for a perspective change. Depending on the hat the participants are currently wearing, they change their way of thinking, i.e., their attitude within the discussion becomes analytical, emotional, critic, optimistic, creative or structuring.

Influence of domain analysis

Domain analysis generates a very specific set of requirements and design rules that need to be followed. The results have a direct influence on the work concerning the requirements analysis. This means that demands from domain-centered engineering have to be merged with functional requirements from the project scope. These can be external influences concerning technical, legal or personal restrictions as well as demands on performance, setup time and reliability, just to name a few.

Provision of existing knowledge

As a third source of demands on functional as well as non-functional requirements, involved project partners bring in their experience from former projects. Certainly, they are specialized with regard to their research field that can be focused on technology but also on the domain. The knowledge that is available right from the project's start is very valuable for the project's engineering and design work. It needs to be considered right from the beginning as basis for future findings.

3.1.2 Iterative Refinement and Extension

Iterative design

Throughout the projects, the consortia have referred favored iterative design approaches as described by Buxton and Sniderman [1980], Nielsen [1993] and Gould and Lewis [1985], predominantly in the scope of user interface (UI) design. Karat [1990] shows that design iterations are expensive and time consuming process steps but still generate economic value.

Prototyping for early feedback

During iterative design, prototyping methods allow for fast constructions of application concepts in terms of hard- and software that can be presented to the users in order to quickly receive feedback on the current design, interaction or application approach. Prototyping methods can range from paper prototypes (cf. Snyder [2003]), to cognitive walkthroughs (cf. Lewis et al. [1990]) and Wizard-of-Oz techniques (cf. Dahlbäck et al. [1993]) up to horizontal or vertical prototypes that either analyze the general concept or deeply focus on central aspects of the functionality. Buchenau and Suri [2000] similarly distinguish between functional and "look like" prototypes that focus on functional depth or provide an overview of the system's behavior.

Iterative refinement

The findings from the prototyping sessions are fed back to design methods and the requirements management tasks. Especially in a research project, requirements need to be formulated and adapted several times since they are not complete at the beginning of the project (cf. Dix et al. [1998]). This knowledge and the one coming from experience must be documented since it is valuable for future designs where it can be formulated as requirements and design challenges for the next iteration.

3.1.3 Dissemination and Exploitation

Research results

The results of the evaluation and validation phases are exploited as research publications and reflect the preparedness of the consortium concerning concrete product developments. According to the generated and validated technical and conceptional knowledge of the research performed, concrete mappings to thorough product demands can be made and implemented.

Future projects

On the other hand, the research consortium has generated a large amount of expert knowledge that enables the partners to apply for follow-up research within a similar domain or to transfer the results in new domains where they concentrate on aspects of their expertise. For the exploitation during and after the development phase it is essential to have documentation of the project's progress and results in a comprehensible and well-accessible form. Lengthy reports and deliverables from which core results need to be extracted in a time-intensive manner are not suited for this purpose.

Internal communication

The same holds true for project internal communication. Different teams and new members can only benefit from reached results if they are described in a transparent and easily accessible way. Latest research on collaboration and information exchange systems combines social media and intranet solutions. The project "expedition enterprise"[4], e.g., analyzes current approaches that are applied within enterprises and identifies needs for internal communication. Similar work is performed by Meyer et al. [2012] and de Rore [2009] in the field of knowledge management for software development and productivity measures within projects.

[4]http://www.expedition-unternehmen.com

3.2 Challenges for Knowledge Distribution

Engineering Open Innovation

At the beginning of a research project , there are already many experiences, processes and methods available that the consortium partners bring in from their individual backgrounds before research starts in the new project's scope. Newly developed processes and approaches need to be described and validated. Beyond this, the findings revealed by validating the concepts and prototypes can be unexpected and surprising which, in principle, is a good fact in research. As a consequence, these results have to be respected and fed back to the overall project knowledge repository. Validation may affect implementation plans for agreed concepts due to discovered change requests.

Difficulties in knowledge management

The following sections discuss the main problems that were encountered in conjunction with information exchange between the work packages domain analysis, interaction design, implementation and integration as well as validation. The focus lies on knowledge *generation*, *exchange* between the project members as well as the integration of *feedback* from application and validation. Figure 3.4 illustrates the influences of the different activities on the overall project knowledge. Direct influences, shown as solid arrows, originate from initial domain analysis, requirements specifications, application and interaction design concepts as well as feedback from implementation and validation efforts. Existing knowledge from the partners and contributions from end user groups additionally affect the common project knowledge. During the iterative design phases of the project, the same initial contributors provide feedback on the generated results and hence directly or indirectly update work results.

3.2.1 Different Sources of Knowledge

End-users' habits and knowledge

A project work structure with mutual interdependencies as described in Section 3.1 and the large amount of involved participants with different backgrounds like, e.g., engineering, computer science, sociology, psychology or economics needs well organized stakeholder management throughout the project's lifetime in order to ensure cooperation and success of the whole endeavor. End-users for whom the system and interaction concepts are designed play an active role during design and evaluation steps. This group possesses essential knowledge that is relevant for the acceptance or rejection of the developed system. Many requirements can be specified directly but some are hidden in the daily workflow or presupposed and therefore unconsciously in the users' minds. Section 3.1.1 discusses these kinds of requirements that can be extended to the concept of domain knowledge in which the specific requirements are embedded. Domain experts represent a whole user group and act as consultants regarding system, interaction and application design as well as technical requirements and given circumstances within the domain.

Partners' specializations

One of the aims of integrated EU funded projects is to bring together research fields and experts from different domains to collaborate and integrate for novel approaches

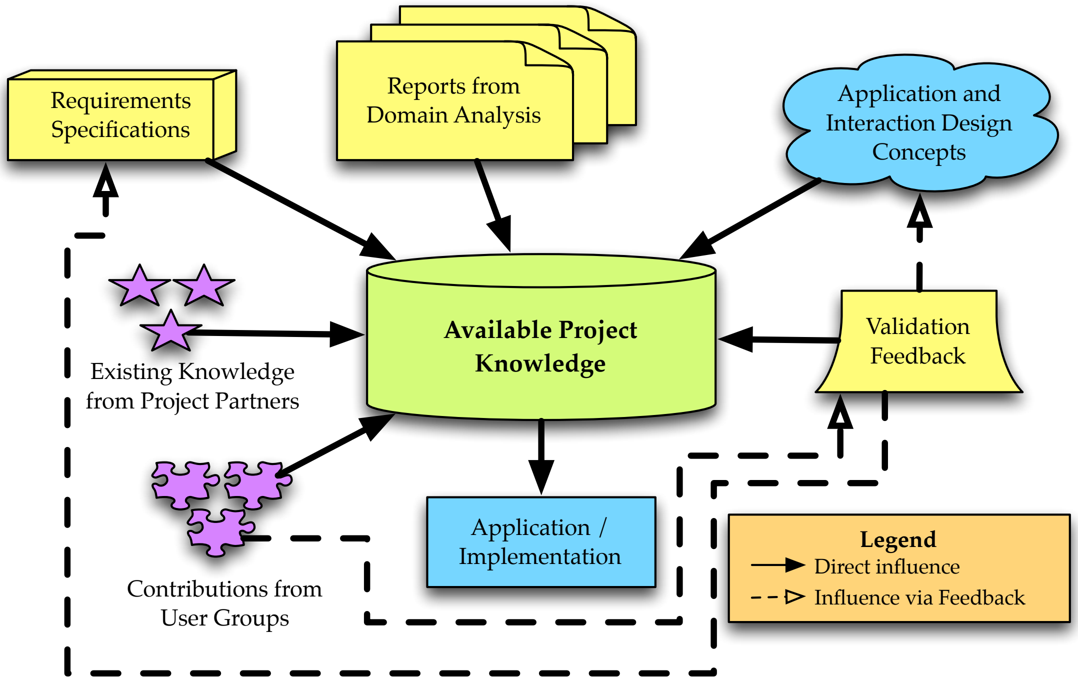


Figure 3.4: Information is introduced from different work packages and different partners right from the beginning of the project.

and combinations. Project partners that are specialized in specific domain-related aspects provide lots of knowledge and experience that they actively need to share and communicate within the project consortium. Sometimes, convincing others of the applied methods is needed. This way, new insights from other former project contexts are brought into the new research topic. Though, not every partner is an absolute expert for the application domain but moreover contributes to the project by bringing in technical solutions or methodical approaches. This results in different states of information about the project partners' work, knowledge and processes, respectively.

Existing mandatory rules

Mandatory rules and guidelines within the project domain have an important impact on the system design and must be considered right from the start. Often, it becomes hard to find documented references regarding successful system design. Domain experts are an essential source of revealing and making aware of these kinds of mandatory requirements. With their experience, important preconditions for the general acceptance of the system concept can be ensured. The same may be the case for more technically-oriented domains in which knowledge about technological possibilities and development procedures or additional restrictions in terms of laws, rules or policies can exist. Processes for validation, evaluation and exploitation also need to follow rules and workflows that may not be clear to other partners. Validation and evaluation knowledge can be continuously reintroduced to the project knowledge repository as collected validation results and feedback from the end user groups.

Knowledge in technical developments

Bailin [1997] states that a portion of knowledge creation that is inherent in software code, system architecture and application design. Software shall not be seen solely as vehicle for automation specification. Dubochet et al. [2009] argue that well-written and designed software code helps in understanding and extracting the domain models within the source code and therefore can be seen as medium for human-to-human communication. According to Didrich and Klein [1996], software products need to be accompanied by up-to-date documentation.

3.2.2 Coordinating Knowledge Transfer

Parallel activities need synchronization

The involvement of many partners and the organization of task groups within work packages lead to the effect that multiple activities are starting concurrently. This means that preparations in all work packages are performed reaching from agreeing on internal processes and milestones to clarifying the technology and infrastructure that already exist. Thus, many activities start together with the detailed studies of the problem domain. The domain analysis produces important reports on given work procedures, technologies and infrastructures. There is also a strong influence on application scenario creation and concept development. All work packages need to incorporate and merge their aims, visions and capabilities to the existing situation. Not only application concepts need to be refined but also applied technology and system designs are affected from validations and feedback that are collected from real situation studies. New requirements may be identified and existing ones may require updating. The processes around project milestones require additional and more widespread communication efforts. Analysis and conception may happen separately and earlier from other efforts that independently start with preparations necessary for their project tasks. According to this, there is the danger of loosing the current state of the partners' activities.

Information overload may lead to ignorance

A well-known problem is that user-centered design methods combined with ethnography tend to produce a large amount of quite lengthy documents (cf. Viller and Sommerville [2000]). Partners with sociology-related backgrounds prefer to present the "large picture". For them, also the reasons and originating story behind the findings are important to record for a more complete understanding of the domain (cf. Garfinkel [1967]). More technical work packages have problems of parsing and digesting all this information. At some point, they even must reduce their efforts on this part and incorporate missing features in the next iteration. The problem of communication between expert groups, especially for ethnography in combination with user-centered design and technicians is discussed in more details by Hughes et al. [1992] and Sommerville et al. [1993]. This may results in unawareness of the individual work package outcomes that are available but not considered by others. There is a lot of self-responsibility given to each work package to updated and distribute gathered knowledge.

Different formats

Due to different methods and processes within the individual work packages, findings are usually reported in different formats. Hence, the results are not always directly reusable as input for other work packages. Domain engineering and interaction design related work packages usually provide reports, drawings, pictures, videos or transcription. Documentation that is well-suited for brainstorming or conveying the imagined concepts are typical for application-design work packages. These can be storyboards, brainstorming maps, story-based scenario or horizontal as well as vertical prototypes. Validation reports from the appropriate work package partially fertilize requirements engineering, implementation and integration efforts. Technically-oriented work packages, however, demand "hard" specifications such that the project management needs to bring together system engineers that approach the domain with a very technical understanding of "what is technically possible" and the user-centered engineering groups that collect information on "how people and processes work" and "what they need".

Knowledge handling and exchange

Domain analysis and requirements engineering activities usually accompany the implementation and validation tasks for more than half of the project runtime. This permits the elaboration of new findings alongside with updating already existing results or collecting experience with assets produced from other work packages in conjunction with the validation processes. All information generated during the domain analysis phase needs to be documented together with the gathered requirements. These documents, that contain specifications and reasons for necessities, are then fed into the appropriate work packages. The same holds true for technical and non-technical documentation from the interaction design and implementation work packages as well as validation reports. All results are accumulated in the general project's knowledge pool from which the appropriate work packages need to extract required knowledge in a format that is suited to their needs. This is the part where most exchange problems result from due to large overhead. Prause et al. [2010b] present results from a survey on research projects within the Sixth EU Framework Programme in which 152 representatives from 73 different projects took part. Extrapolation in total regarded 741 projects out of 1167 as relevant for the study. The evaluation revealed that lots of communication took place within the projects, but predominantly via emails, phone calls and chats. Exchanged documentation was only attached without preparation for the recipient. Interested parties need to actively transform the formats to their needs.

Transformation and adaptation

Thus, in order to exchange the generated knowledge and reuse it for the specific tasks, transformations are necessary. The cores of the reports and stories must be transformed into requirements that can be implemented and validated afterwards. From resulting user tests in an iterative design approach, feedback is formulated in the work package specific documentation and then needs to be reformulated again. Meyer et al. [2012] describe similar phenomena that they experienced in a joint project with research and industrial partners. Whereas research-oriented partners tend to provide more abstract information that serves the exchange of experiences and derivation of new concepts, industry partners preferred stricter guidelines. They found in a case study that it is hard to fit new workflows into existing procedures. In combination with technical issues like system availability and response time, an introduced knowledge management process is strongly endangered to fail.

Knowledge changes

Later in the project, when parts are being implemented and first prototypes are validated, findings also need to be recorded and documented. Again, there are many different kinds of documentations possible. The problem here is to feed the results back into the project and appropriately inform the other project members. Additionally, the later new associates enter the project, the more the information flood increases. New requirements may first come to mind when the system or prototype is currently used (cf. Dix et al. [1998]). This knowledge and the one coming from experience must also be documented since it is valuable for future designs.

Feedback from validation

Throughout the different phases of engineering within the project, many end users are interviewed and being included in design and conception tasks, e.g. during surveys and user workshops. The different outcomes need to be synchronized with the overall project conceptions and eventually aligned with the results from the validation phases. Every work package must be aware of the updates and eventual changes. The communication overhead must remain low for all participants by, e.g., avoiding transformations of the work package outputs.

3.2.3 Reasons for Insufficient Knowledge Transfer

Time, roles and motivations

The information exchange at the synchronization points for the efforts are manifold. A small-scale survey among project participants (cf. Section 4.2.2) revealed that documentation itself creates overhead for information exchange. Many efforts need to be put in adapting contents to specific needs. Additionally, time restrictions often hinder the creation or reading documentation. Another concern among the interviewees was uncertainty about the activities of others, i.e., they were worried to be the only ones creating documentation or that their contributions will not be regarded by others. In some cases, the same project members from one partner have tasks in different work packages. Therefore, they do not regard the documentation of findings as necessary. Only if they really have to document their findings because of mandatory reports or specifications, the additional work steps are undertaken. There is a motivation problem of assembling detailed descriptions.

Loss of important lines of thought

Details about solutions or reasons for or against decisions, respectively, are forgotten over time and cannot be restored from memory. This means, that eventually important argumentation or explanation are lost for the group. The same effect may happen in case participants leave the project and take their assembled knowledge with them. To some degree, sharing and relaying take place in handover documents but still a lot of tacit special knowledge is lost that cannot explicitly communicated via written reports.

Volatile knowledge

Meyer et al. [2012] explain reasons for low motivation regarding the creation of documentation that result from the volatile character of information. Information may age faster than it can be reported. In combination with latency on updates such that the retrieved information does not reflect the current state, the motivation drops even worse.

This may lead to not taking into account the documentation anymore and therefore hindering the information exchange.

Hidden agenda

Another potential for hindering information creation and exchange is an eventually existing unwillingness of sharing results. Hidden agendas of the participating project partners may be a reason such as custom exploitation and production plan. Although this conflicts with research projects ethics, this possibility should be listed as a threat.

3.3 Concept Validation in the Emergency Response Domain

Integrated research project over 4 years

The engineering and validation processes for the approach presented in this work are embedded in the scope of the BRIDGE project that started in April 2011 and runs until March 2015. The project is funded by the European Union Seventh Framework Programme within the Security Programme SEC-2010.4.2-1; Interoperability of data, systems, tools and equipment. BRIDGE provides the characteristics of an international, distributed integrated research project (IP) with different sources of knowledge generation and specializations in many different technical and non-technical levels. The total project costs span €18,075,144.20 with a EU contribution of 71.8% and started 1 April 2011 with a duration of 48 months.

Project aims and structure

In total, around 60 researchers and developers working in 14 institutions located in 7 different countries seek for solutions to support crisis management and technical interoperability in large-scale emergencies. Possible scenarios cover disasters caused by technical failures or natural catastrophes as well as terrorist threats. The treated questions range from integrating different information sources from the different units in the field over the incorporation of agent-based systems and process analysis up to the preparation of 3D model-based simulations. The professional backgrounds of the involved staff range from academics, technology developers, domain experts to end-user representatives. take part in the project activities. BRIDGE is structured into 13 work packages (WP) as illustrated in Figure 3.5. At the top, work package 12 is concerned with social, ethical and legal aspects. Work packages 2, 4, 8 and 10 explore the project's problem domain and handle integration as well as validation tasks. They surround the application-related activities of work packages 3, 5, 6 and 7 that are shown as inner boxes. Activities related to project-management, representation, dissemination and exploitation are handled by work packages 1, 9, 11 and 13 at the bottom. The work packages have different strengths of mutual influence. Especially the work packages concerned with domain analysis (WP2) and interaction design (WP6) need to fertilize each other. Furthermore, WP6 is strongly depends on feedback from WP2 and WP10 that validate developed approaches. Others, however, individually accomplish their tasks for a long time on their own until demonstration (WP9) and integration efforts (WP8) need to be made. The same holds true for dissemination (WP13) and exploitation activities (WP11) that need current state information at discrete points in time.

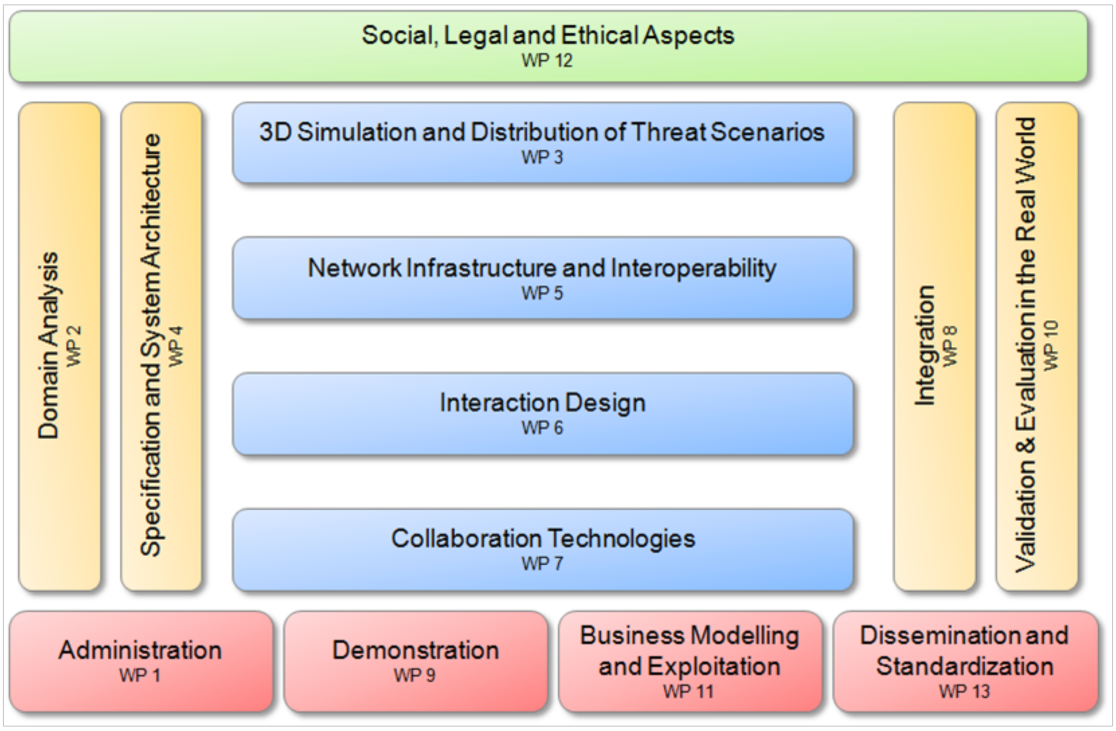


Figure 3.5: Overview of the BRIDGE WP structure.[5]

3.3.1 Peculiarities

Tacit knowledge in practice

Engineering assistive and collaborative systems in the domain of emergency response revealed that there is lots of tacit knowledge in the minds of the end-users which are predominantly medics, firefighters, policemen and incident commanders. Since the problem scope of the project considers inter-agency and cross-border collaboration, many different regulations from different countries are shuffled. Stakeholders were directly involved to document their knowledge since participatory design cannot be applied in all situations due to the dangerous nature of the training and real emergency cases with fire and injuries. However, in order to be able to create working and accepted concepts, applications and hardware designs, many pre-existing conditions, rules, processes and behaviors must be understood by the designer. Otherwise, the proposal of new technology or concepts will directly be rejected by the target groups.

Critical against new technology

Domain analysis within the project shows that there are already settled and well defined processes and technologies for safety- and time critical emergency situations. This means that the tolerance for design and process changes is very narrow. As the design premises formulated by Turoff et al. [2004] suggest: "An emergency system that is not used on a regular basis before an emergency will never be of use in an actual emer-

[5]Source: http://www.bridgeproject.eu/en/about-bridge/project-structure

gency." (cf. Premise 1, p.6). The target user group that consists of emergency response units in the field as well as incident commanders are experts in their domain with a lot of experience and best practices to apply. The information base on which decisions must be made is not always appropriate: Poor availability of information or flooding leading to mental overload may occur (cf. Carver and Turoff [2007]). First responders need to be sure about the reliability of the systems and must be able to automatically apply hardware and operate tools and gadgets. Novel technologies and hardware represent new sources of dangers that may distract or hinder established processes.

Requirements per user group

Every user group has its own requirements for a system and already experience that must be considered during system design. There is a huge gap between system engineers that approach the domain with a very technical vision of potentials and the user groups that needs to act in dangerous and hectic situations and therefore do not want to learn yet another technology that eventually even disturbs them in the tasks they are currently pursuing. Lots of training sessions are needed in order to ensure automated interaction with the used systems and equipment. It is hard to change or integrate new technology into established processes except for justified and convincing reasons. Changes must reliably bring improvements over the existing tools and processes. The target user group is not to be considered as narrow-minded but reasonably critic towards innovation that may in a worst case scenario endanger victims and responders.

New approaches needed for HCI

Therefore, system and application design within the emergency response domain require special care. End users might be in a stressful situation where lives are at stake. Traditional human-computer-interaction (HCI) approaches can turn out as inappropriate. Therefore, new approaches must be created that simplify the interaction with the new system design. New interaction models must support the user and help him achieve a natural interaction with the device. Especially when interacting with emergency management systems, people must feel confident with the system they use under the given conditions. It is often necessary to adapt existing validated design concepts from the domain since most systems in emergency response are based on an easy-to-use graphical-user-interfaces (GUI) with special attention to the needs of the user. General, existing design patterns are not necessarily applicable for safety critical environments and emergency management due to their general description and usage. In order to be applicable, design concepts or patterns must be translated and set into the proper context. Furthermore, they must be collected in a structured manner and validated periodically based on progress in the current domain. Most online pattern collections contain general design patterns for user interface and interaction design. For the application in ER scenarios, the patterns must be carefully incorporated and revalidated.

Durable hardware

Hardware design is another very important factor. The devices or sensors that intend to support the first responders must be very durable, i.e. water-proof or heat-resistant, and be usable with the current equipment the user wears and the situation in which he normally has to react. This can be rubber gloves in the case of medics, or visors hindering the sight of firefighters. New functionality has to be embedded into existing equipment and must not interfere with applied procedures.

3.3.2 Existing Patterns

Survey on ER patterns

Project research efforts during the domain exploration and technical conception phases revealed existing patterns for the design of emergency response systems. In the following, a survey on relevant pattern collections and languages is presented together with an overview of related approaches that serve as knowledge source from which new patterns can be formulated.

PL for non-documented processes in firefighting units

Denef [2011] presents a pattern language describing the social structures within firefighting units. Non-standardized but still often applied processes on duty, self-made tools and courses of action are described that are usually not documented officially but learned at work. Such patterns help in understanding the domain, its processes and stakeholders. Thus, the collected knowledge is relevant for application and interaction design since it also provides understanding for the users' behavior and what kind of newly developed tools or device they likely will apply or reject.

ER management

The work of Acuna et al. [2010] describes the development of a design catalog for web-based emergency management . This design catalog is created based on the different phases of a disaster with respect to first responders and the command control stands: Preparedness, Emergency Response, Rehabilitation, and Mitigation. Therefore, different already existing platforms are evaluated to get an overview of the most important activities within the phases. The study resulted in 19 design patterns grouped into 4 sections, namely (i) Emergency Response and Recovery, (ii) Planning and Mitigation, (iii) Information Management and (iv) Access for all.

Process for pattern development and integration

Grill and Blauhut [2008] describe the consequent application of design patterns as a step in the software system engineering process. The *Frequentis* systems engineering process for safety-critical environments (SCE) integrates the application of user interface design patterns as well as their continuous validation and update. The described framework identifies and validates design patterns within the field of emergency management through a user-centered design approach. The framework extends the user interface design process by introducing periodic evaluations of the pattern validity. This results in an company-internal up-to-date UI design pattern library containing the experiences of different designs in safety-critical environments.

Patterns for safe human-machine interface

A basic set of patterns that are intended to support the development of safe human-machine interfaces within the same domain is described by Hussey [1999]. He takes into account usability aspects that play an important role for avoiding human error and thus retaining the whole system in a stable and safe state. The patterns are formulated around the "safety first" paradigm. Hussey formulates 8 patterns including guidance for collection that covers task management and execution, presentation of information as well as machine control. The concepts were evaluated in real designs and represent important basic rules that can be applied for system design in the domain of emergency response. Connelly et al. [2001] extend this collection with custom patterns and further synthesize user interfaces from the pattern collection. The presented case study supports and demonstrates their quality and validity. Concerning the discovery of new patterns, the authors state that this task is not trivial. The application of existing patterns is described as time consuming.

3.3.3 Related Approaches as Pattern Sources for Pattern Formulation

Pattern candidates

During the research on related approaches, more general concepts on human-computer-interaction in safety-critical environments were discovered that are not formulated as patterns. Due to their relation to emergency response, they are considered as relevant background knowledge for system design and candidates for design patterns.

Symbolic representations

Newlove-Eriksson and Hermansson [2010] provide a detailed overview of the symbolic representations. The signs or icons act as a marker for further information. For example, by clicking on this marker a video is displayed or an image is presented. This reduces the complexity for the user and facilitates a better overview of the situation, especially on small devices. Icons can also be used for interaction between different team members. Fitrianie and Leon [2007] give an example of an icon language.

Task management

Based on the assumption that humans can only process 4 items simultaneously, Humayoun et al. [2009] suggest that the screen area should be divided into four sections, where each of the four sections should be colorized depending on the displayed content. An example of this is given by the Task-Handler of the *WORKPAD* project, which is a user interface designed to support the overview of teams and their assigned tasks. The *WORKPAD* application may be a potential source for deriving more patterns for the area of emergency response, especially for task assignment or handling.

Gesture handling

Methods for handling gestures are very common in the area of disaster management. The approach of Bader et al. [2008] describes a disaster management system with a tabletop display where a map of the affected area is displayed. Due to cameras above the display, gestures from the hand can be recognized. This enables the user to perform actions such as selecting, zooming, and similar, by pointing at or sweeping over the appropriate area. Besides the gesture recognition, the system also facilitates intra- and inter-team communication.

Visualizations for shared situation awareness

Hanratty et al. [2009] emphasize the importance of shared situation awareness in human-agent networks. This can be transferred to the situation awareness in critical situations, where clever visualization techniques are needed to get a better insight into the complex situations. Their approach visualizes human beings and software agents in past and currently evolving situations. This supports users to understand complex situations and to make decisions.

Mobile triage

Nestler and Klinker [2007] describe a mobile user-interface for disaster management and health care personnel performing triage[6]. At emergency sites, patients are assigned

[6]Triage: The process for sorting injured people into groups based on their need for or likely benefit from immediate medical treatment (see http://www.thefreedictionary.com/triage).

to more than one relief team and vice versa. Hence, three different representations are needed to give an overview for all relief units: (i) Static knowledge (treatment, coordination procedure etc.), (ii) patient related information (particular history etc.) and (iii) peripheral information (number, states and position of patients and relief workers etc.). To obtain an overview of the patients within an area, the authors make use of a map with standardized symbols for marking the position of patients, their state and number, and also information about relief units. Detailed information about a patient can be requested from the map, where information about the patient's condition is included.

Ad hoc collaboration

The research efforts of Monares et al. [2011] describe a mobile application for firefighters called MobileMap. It comprises a low-budget, easy-to-use system for fire departments. With the system it should be possible to overcome some of the problems concerning the communication during an emergency (e.g. interfering radio channels, limited amount of radio channels, communication overload within channels). To achieve this, the system supports ad hoc communication and collaboration among firefighters, and exchange of rich information to facilitate decision making. The user interface facilitates a map in which the user can choose from different context data and resources.

Support by video

Multimedia data within disaster management is an important source to gain as much information as possible about the situation. The work of Bergstrand and Landgren [2009] shows that video is a very important source for information, and that it can be a positive contribution to the work practice.

3.4 Conclusion

Standards are hard to apply

Standardized project management processes and rules cannot directly be applied in the landscape of distributed research project. Reasons for this are primarily the artificial project organization into work packages suggesting a separation of concerns and tasks. Additionally, the project partners still keep a certain degree of self-determination for their tasks and therefore may evade obligations.

Problems for information exchange

One effect of this project structure is, even in a cooperative project atmosphere, a decrease in communication and exchange. Latencies in development originate from constraints within a work package resources. In case that results from other work packages cannot easily be integrated into the current work, important synergies may get lost. Knowledge generation within such a project structure is manifold and communication means are different. In addition to this, the consortium partners need lots of time to read and understand all documented development steps and results. They demand condensed summaries of the results that are adapted to their needs.

Different formats and magnitudes

Along with different engineering processes between the project partners the documentation formats may differ a lot. Technology-driven work packages tend to produce short and specific documentation with class diagrams, architectural views or class diagrams that are suited to software experts' needs. Other work packages that are more involved in socio-technical tasks such as domain engineering and user-centered design produce scenarios written in long prose, storyboards or transcriptions of interviews as well as hours of video recordings. For their work, the reasons and originating story behind the findings are important to record and understand. However, the produced material may not be suited as input to other work packages or not understandable for other partners.

Synchronizing different sources of knowledge

Knowledge originates from end-user demands and resides within their processes. Additionally, partners within the consortium bring pre-existing knowledge into the project. Mandatory rules within a domain and knowledge that is encapsulated in software and hardware artifacts represent another source of knowledge. All efforts need to be coordinated between the work packages. This may lead to a synchronization problems regarding the efforts undertaken by each partner.

Reasons for lacking knowledge transfer

Reasons for incomplete knowledge transfers between the work packages may result from a lack of motivation to actually document their knowledge. This can be due to time restrictions or the unawareness of the activities of others. It is also possible that there currently is no real recipient except of the same person since the project member has multiple roles in multiple work package activities. Additionally, knowledge may be seen as volatile such that the documented information is outdated too quickly. Hidden agendas as a very unethical behavior within a research project are mentioned as a possible reason for refraining information.

The BRIDGE project as testbed

As accompanying engineering example, the BRIDGE project is introduced and used as testbed throughout this thesis. The project aims at improving inter-agency and cross-border collaboration in emergency response. The peculiarities of the project domain regarding design methods, user attitudes towards new technologies and hardware designs, place new demands on application and system design and need adaptions of the conventional design processes. Existing pattern collections and research work related to the ER domain that is seen potential source for further patterns were briefly presented in this context.

Identification of missing features

The following chapter presents the research agenda pursued for engineering the approach of this thesis. Based on experiences gathered in different workshops, problems from a feature comparison between existing pattern languages and mining approaches are extracted. From the workshop results, requirements for the conception of a collaborative pattern-supported engineering process are formulated.

Chapter 4

Towards a Collaborative Formulation Process for Evolving Patterns

Patterns for mutual communication

The approach of this thesis makes use of patterns as interdisciplinary communication medium, especially for describing and explaining the current application domain, its characteristics and derived concepts. The aim is to improve the mutual understanding and communication between work packages especially concerned with domain analysis, design and validation. In addition, all project members benefit from the descriptions so that, even if their work is very technology-oriented, they have an easier access to the gathered information and derived concepts. The understanding regarding *why* a specific concept has to be implemented in the given way can be established. The envisioned approach explicitly involves more stakeholders in the knowledge engineering process besides the work packages' experts.

Organization in an evolving pattern library

Patterns that undergo the collaborative formulation and refinement process are kept within a dynamically changing *evolving pattern library* (EPL) that reflects the development of the patterns themselves but also the structural dependencies between them. The discussions on current problems assess existing approaches of organizing patterns and outline missing qualities with regard to the pattern usage as communication medium within a research project. The collected requirements take into account the need for a collaborative pattern library structure as pattern management and formulation platform.

Cooperation with Reutlingen University

The concept of patterns as parts of knowledge and means of communication was widely accepted by the BRIDGE project consortium. In parallel, an exchange of experiences took place with the Architecture Reference Lab (ARL) that is lead by Professor Dr. Alfred Zimmermann at Reutlingen University[1]. The contact could be established during

[1]http://reutlingen-university.de

pattern-related conferences. This resulted in a mutual exchange of experiences with regard to pattern formulation and validation in the individual project domains. For the ARL Group, the focus lies on Service Oriented Architectures (SOA) within enterprise software architectures. In particular, the knowledge that is generated within research and industry projects is regularly collected as patterns in the reduced canonical form within pattern catalogs as described by Zimmermann et al. [2011]. Common considerations revealed that an iterative and collaborative pattern formulation process was considered as promising candidate for documenting the knowledge generation and refinement process as outlined in Reiners et al. [2014b], Zimmermann and Reiners [2012].

Similar problems in different domains

In both groups, the currently applied derivation processes for patterns were considered as cumbersome and time consuming. These concerns are addressed by introducing collaborative components into the pattern formulation process. This way, findings should continuously be documented as patterns throughout the whole project lifetime and so iteratively improve and validate the developed solutions. The current state of findings and concepts is reflected and potentially every project member can actively take part in the discussion and validation process. Thus, the work of the individual work packages should become clearer to others by monitoring the pattern formulation activities.

From problems to requirements

Section 4.1 describes the engineering efforts that were undertaken to derive the idea of collaboratively formulating patterns and their iterative refinement and validation. The conducted expert workshops within the research groups with the aim to identify missing features and shortcomings of the current methods with regard to the realization of the envisioned approach are described by Section 4.2. Based on the analysis, the subsequent Section 4.3 discusses the initial requirements on the approach that build the foundation for the collaborative pattern formulation and maturation process that is described in Chapter 5.

4.1 Research Agenda

Problem taken up from existing projects

First ideas of the problems were encountered by the author during active participation in research projects or from an external perspective as observer during regular progress meetings (cf. the overview of the project participation in Section 3.1). The research agenda as presented in Figure 4.1 shows the author's engineering and validation efforts for the presented evolving pattern library approach. For its conception, several expert workshops were conducted in existing project environments. Based on conceptual walkthroughs and interactive prototypes, the approach was refined and finalized in an iterative and user-centered manner, shown as incremental steps in a spiral shape similar to the according model proposed by Boehm [1986]. The illustration describes three complete iteration cycles whereas the last process step intentionally exceeds the illustration in order to indicate the opportunity for future iterations that are beyond the scope of this thesis.

The research efforts started with the general idea of using design patterns as micro documentation for improving the documentation and information exchange processes within research projects. With the start of the BRIDGE project in April 2011, the approach was embedded into the project and an expert workshop with representatives from the work packages of domain analysis and interaction design was held in September 2011 in order to identify a first set of needed features for the envisioned approach. Additionally, existing pattern language concepts were analyzed and compared with demands that were formulated for the envisioned platform. Missing features that hindered the direct usage of the standard pattern concepts were identified and first solutions were proposed. In parallel, a workshop on current problems for formulating and managing patterns during the project's lifetime was held with pattern experts from the SOA domain at the ARL Group at Reutlingen University in November 2011.

Initial workshop on missing features

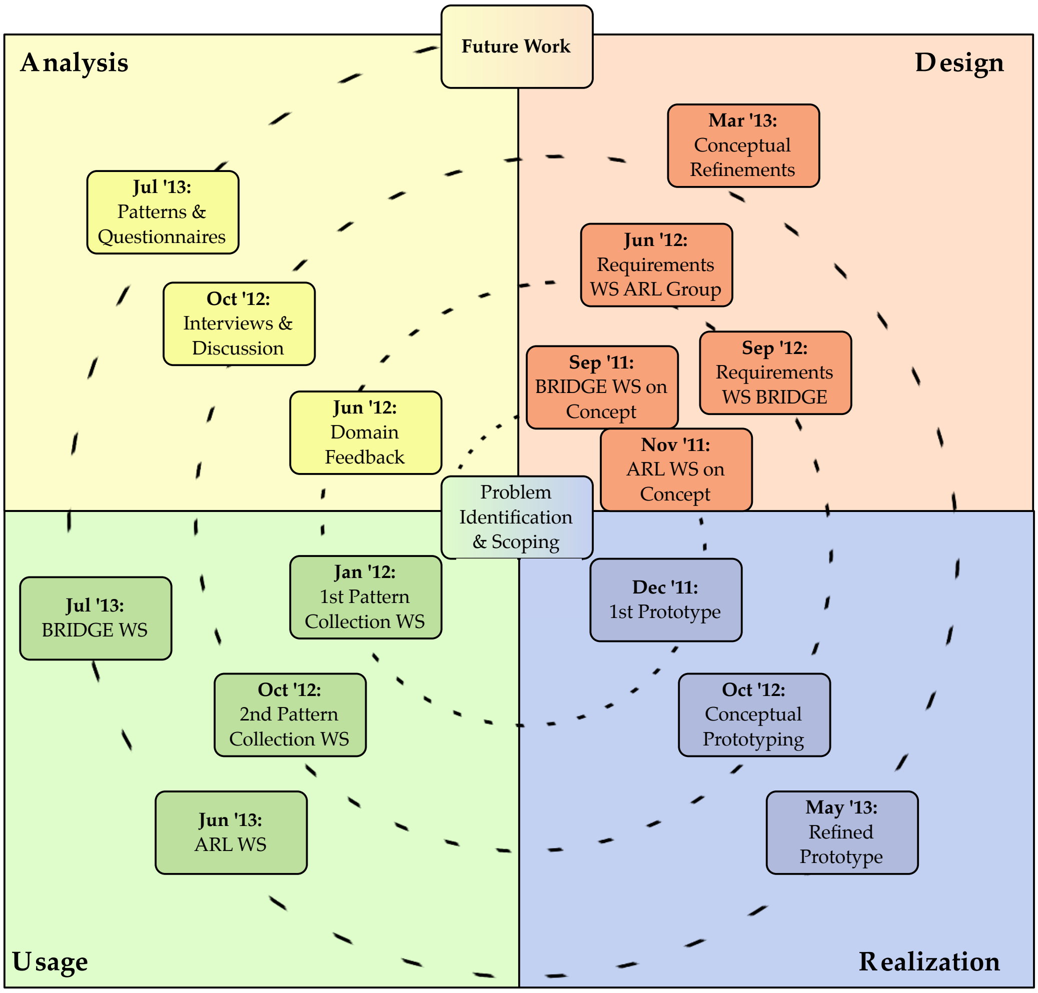


Figure 4.1: The iterative development steps of the pattern library approach.

Pattern mining for prototype

An initial set of needed features that were identified during the first conceptual workshops found its way into the BRIDGE Pattern Library prototype[2] that was published in December 2011 (cf. Section 4.3). In order to fill the initial design pattern library prototype, a pattern collection workshop was conducted with members that were predominantly involved in the domain analysis and interaction design work packages. The participants were invited to remotely submit patterns to the library and were asked to formulate an initial set of patterns that originated from their current results and visions of a possible realization of the system. In a usage period starting from the pattern collection workshop until June 2012, they provided qualitative feedback, suggestions on the pattern contribution process, the library visualization and means of browsing the pattern library. An initial set of patterns could be gathered during that period. The concepts together with the lessons learned so far were formulated in Reiners et al. [2014b] and elaborately discussed and assessed in a pattern expert round during a Writers' Workshop at the 19th Conference on Pattern Languages of Programs (PLoP) 2012[3].

2nd iteration on more elaborated concepts

In addition to the feedback gathered so far, two further expert workshops were held in June and September 2012 with pattern experts from the SOA domain at Reutlingen University and BRIDGE project members from the work packages concerning domain analysis, interaction design and validation, respectively. Figure 4.2 shows impressions from the discussions and experience exchange. Both workshops revealed further requirements and suggestions for improving the concept that could directly be based on concept presentations in combination with the first online prototype. The extended requirements and suggestions for improvements were further analyzed with conceptual walkthroughs and personal interviews until October 2012. Additionally, new ways to visualize the design pattern library structure were derived and discussed with the help of paper prototypes. These workshops were directly aligned with a second pattern formulation workshop that helped to understand the newly derived concepts. The formulated patterns were included in the online prototype to further support its development based on the new conceptions.

Consolidation and relaunch

In March 2013, the feedback from the different workshops and interviews was consolidated and resulted in an advanced concept for an evolving design pattern library as well as a new visualization concept of its structure and liveliness. The relaunch of the pattern library prototype that included all refined concepts of the second iteration took place in June 2013. A modified version of the prototype for a summer school workshop held at Reutlingen University was prepared for conducting an acceptance study for the pattern maturation process and the visualization mechanisms. In the scope of the project BRIDGE, a third pattern collection and feedback workshop was organized in July 2013 in which the opinions about the new visualization and interaction concepts were collected and prepared for the formulation of future work.

[2] http://pattern-library.sec-bridge.eu
[3] http://www.hillside.net/plop/2012

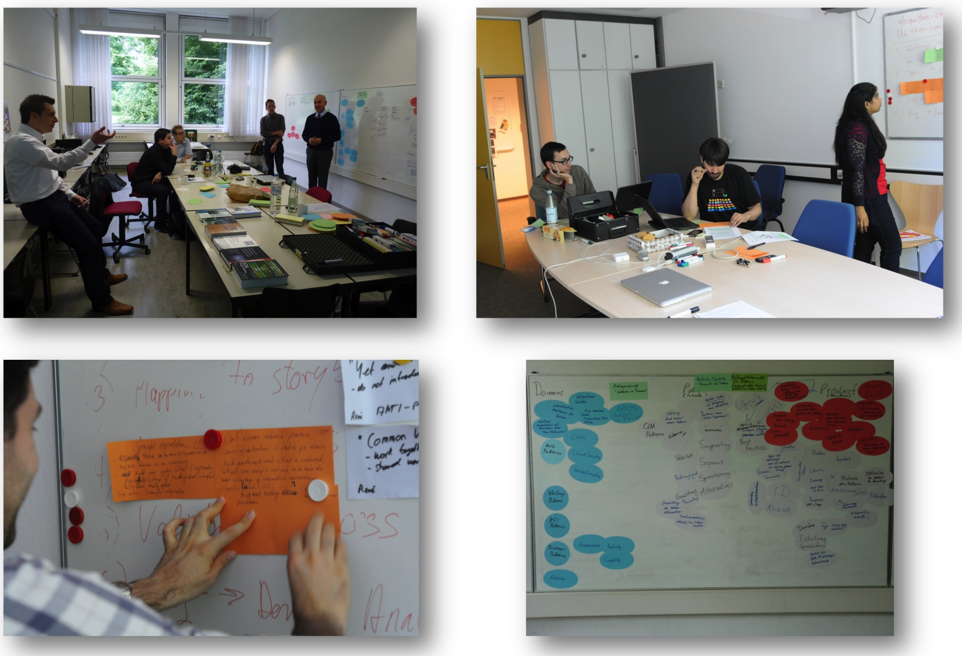


Figure 4.2: Impressions of the workshops held at Reutlingen University in June 2012 and Fraunhofer FIT in September 2012, respectively.

4.2 Missing Features of Existing Processes

Organization of workshops

During the expert workshops for the concept derivations of the first iteration of the BRIDGE design pattern library in September and December 2011 as well as the refinement workshops in June and September 2012, prominent pattern collection and formulation approaches as described in Chapter 2 were taken into account. The participants of the BRIDGE project were involved in work packages concerned with domain analysis, interaction design and validation, respectively. The second workshop was conducted at Reutlingen University with experts from the SOA domain. Both expert groups were familiar with most of the discussed pattern languages and collections. Access to the printed media and online portals was provided to assess these details during the workshop. For each workshop, eight project representatives could be recruited. Both workshops revealed requirements and suggestions for the envisioned concept of a collaborative pattern formulation process embedded into a dynamically developing repository. As basis for the workshop discussions, prominent pattern collections and languages as described in Sections 2.2.1 and 2.2 were taken into account. For each approach, the experts within the workshop were asked to analyze the domains, the target groups as well as the mining process described by the authors. Then, the experts were asked to rate on different features of the approaches based on a Likert scale covering the values -2, -1, 0, +1 and +2 where -2 meant that the feature was not covered at all and +2 stood for full coverage of the feature. The regarded features covered collaboration and reader's influence on the pattern formulations, extensibility of the pattern collections or language, respectively, with new contributions, the reusability of the formulated patterns as well as the transparency of the mining process for readers. Additional concepts such as the inclusion of bad practices, recommendation of patterns or pattern sequences as well as overall guidance through the pattern collection or language, respectively, were rated.

Discussion of results during PLoP conferences

The following sections describe the results on current pattern structure features concerning collaboration and extension of existing structures together with mining approaches that are in use. The summarized explanations reflect collected opinions from the workshop participants. During preceding work, similar problems were identified when applying the concept of pattern languages to ubiquitous application design and knowledge documentation in distributed research projects (cf. Reiners [2010], Reiners et al. [2014b]). The findings of the workshops were prepared in Reiners et al. [2014a] and discussed during a writer's workshop at the EuroPLop 2013 conference[4] where it served as platform for further discussing the evolving pattern library concept and missing features of currently used approaches.

4.2.1 Feature Comparison

Comparison of a representative set

A comparison of existing pattern languages that are published as books and, in a more dynamic way, web portals showed that current pattern mining and formulation processes still face a number of challenges that need to be addressed in the scope of the envisioned pattern-engineering approach. The results of the comparison that is based on the research of a representative state of the art as described in Chapter 2 are shown in Table 4.1. The table compares the field of pattern collections, in which structural and hierarchical relations between the patterns are rather neglected, and pattern languages that also consider hierarchies and relations between patterns.

Printed vs. web-based pattern publications

Both kinds of pattern organization structures, i.e., pattern collections and pattern languages, were traditionally published in a printed format. Pattern formulation often happens during after a long period of research conducted by the authors. With the growing importance of the World-Wide Web in the late 90s and together with online collaboration mechanism that were developed on top of the Web infrastructure, online available pattern portals were realized. This way, quick publications of patterns and user-based feedback mechanisms became possible.

Evaluation

Table 4.1 summarizes the analyzed domains in which pattern collections and pattern languages were found during the workshops. The comparison underlines the interdisciplinary aspect of the pattern approach since it is applied in many different domains. Usually, they are written in a comprehensive way that also non-experts or novices in the field are able to benefit from the described concepts. Understandability is even more

[4]http://www.europlop.net

	Pattern Collections & Pattern Languages	
	Printed	**Online**
Domains	Software Engineering, Emergency Response, User Interface Design, Website Design	Architecture, HCI, Processes, Education, Emergency Response, Website Design
Target Group	Professionals, Domain Experts, Novices	Experts, Novices, Interested parties $\Rightarrow$ Focus on knowledge sharing
Structure	Clusters by purpose, behavior, category, topic, development step	Hierarchies and dependencies on spatial or temporal aspects, implementation detail, task order
Collaboration and Influence	$\ominus\ominus$	$\odot$
Extensibility	$\ominus\ominus$	$\oplus$
Reusability	$\odot$	$\odot$
Transparency	$\ominus\ominus$	$\odot$
Bad Practices	$\ominus\ominus$	$\ominus$
Recommendation	$\odot$	$\oplus$
Guidance	$\oplus$	$\oplus$
Mining Process	(Internal) assessment, mining workshops, partially shared on conferences $\Rightarrow$ Based on experience.	Online discussion based on commenting, rating and voting $\Rightarrow$ Community-based proposals.

Table 4.1: A comparison and assessment of relevant features provided by the analyzed pattern collections and patterns languages.

enforced in pattern languages that are structured by hierarchies and interconnections between the patterns. This allow authors to structure the knowledge and guide readers through the information. In case that specific concepts must be understood before reading the current pattern, the hierarchy allows readers to see which patterns need to be considered first. This way, the pattern organization structure reflects the recommended order of reading. From a general point of view, the workshop participants requested necessary features to realize a collaborative pattern formulation approach and using them as communication medium. Current approaches were considered as not fully supporting the needs. The different symbols $\ominus\ominus$, $\ominus$, $\odot$, $\oplus$ and $\oplus\oplus$ reflect the average perception of features that were examined within the analyzed approaches. $\ominus\ominus$ stands for no support of the considered feature, and $\ominus$ for rudimentary support. $\odot$ is given for features that are present but do not seem mature enough and were maybe not focused at development time of the pattern collection or language, respectively. ⊕ and ⊕⊕ are attributed for features that are strongly supported or in an outstanding way, respectively.

Collaboration and influence on pattern formulation

Collaboration features of the analyzed approaches were seen as very critic. Interested parties can rarely influence the derivation of the patterns that are formulated for printed publications. Only *after* publishing of the patterns, authors can collect feedback from the readership and eventually incorporate it into the formulation. Knowledge and opinions can only be incorporated rarely during the pattern formulation and solution seeking process. Only web-based approaches allow others to make suggestions on improving the formulated patterns. The analyzed approaches provided feedback mechanisms on the basis of comments and voting mechanisms that only allow indirect influences on the pattern formulation. Workshops were seen as not suited as external influence since only a selected group of interested people can take influence during the formulation process of the pattern.

Extensibility

Once a pattern collection or language, respectively, is formulated and published, only web-based approaches allow for perceivable extensions of the formulated patterns. Printed media can only be updated after a repeated editorial process together via new editions. Web-based processes clearly have advantages in this aspect as stated by the workshop participants. Still, the problem of keeping users motivated and active was seen as most important.

Reusability

Another factor that was analyzed referred to the reusability of formulated knowledge in a different domain in which similar design challenges may occur. In the scope of the BRIDGE project, examples were seen in which interaction and user interface designs from existing patterns could partially be included. Special modifications were still necessary but the external design patterns could be reused as a basis. Thus, reusability in a cross-domain manner can be addressed. So far, none of the present approaches was considered as providing such concepts.

Transparency

The transparency of the pattern mining process, the formulation of solutions as well as the rating of patterns and the validity of the suggested solutions were also evaluated by the workshop participants. Certainly, a very important fact on the validity of a pattern solution is the author group's reputation and therewith the trust in its formulations. For the collaborative approach that is envisioned in research projects that aims at formulating findings and ideas from the very beginning, none of the present approach respects means of validity of the derivation of pattern. The participants were convinced that this has not yet been necessary due to the nature of the existing approaches. Web-based pattern collections and languages, however, have a slight advantage in this regard. User-based rating by votes and a pattern's acceptance and rejection can be understood by comparing the votes and reading the comments.

Bad Practices

In order to fertilize the development process within a research project, early ideas and conceptions should be included. The envisioned approach supports the introduction and refinement of concepts as early pattern ideas that are validated in parallel to the development process. Therefore, it is important to document unsuccessful concepts whose derivation is also possible in an open research field. In order to learn from mistakes and avoid them in future, improved iterations, design failures that can be formulated as *anti-patterns*, are regarded as important to be documented together with successful approaches. Many current publications mostly concentrate on successful approaches and keep silent about failures.

Recommendation

Especially the printed pattern languages by Alexander [1977] and Borchers [2001] give recommendations on applying patterns. This is done by ratings that are based on the authors' experiences. This way, design alternatives or combinations of patterns can be recommended to readers helping them with deciding for or against provided patterns. Online approaches often generate average rankings based on voting from site visitors. This way, many external opinions and experience can be collected and reflected in an individual pattern's rating. The participants of the workshop regarded this community-based approach as stronger support over the ratings of a smaller set of authors. In their view, authors may tend to a positive bias in favor of the own patterns because of latent goodwill.

Guidance

Closely combined with recommendation, guidance is provided by linking to follow-up patterns. Most pattern collections and languages give advice to read dedicated sets of patterns first in order to understand the topics treated by later patterns. At the end of each pattern formulation, authors often give references to related patterns deepening the current topic. This feature was found in printed as well as online publications.

Mining processes

The last aspect that was regarded during the workshops examined the ways how patterns were derived over time. From the survey, the approaches for printed publications were considered as similar. The same was stated for online approaches. This means that pattern derivation mostly takes place internally among a small author group first. Later on, discussions in dedicated workshops follow leading to improved versions of the patterns. Validation is mostly based on the author group's or workshop participant' experience. Iba and Isaku [2014] present the approach of directly involving many contributors at the same time in large-scale and time-intensive workshops (cf. Section 2.3.2). Online approaches provide mechanisms to dynamically introduce new patterns that are assessed by the online community. Based on feedback as comments and votes, the validation and rating of a proposed pattern is determined. The latter approaches were seen as more dynamic compared to printed publications.

4.2.2 Derived Problems From a Feature Comparison

Extracted problems

From the assessment of currently available approaches, a collection of problems that were seen from current approaches was extracted . The derived problems represent a summary of the stated opinions by the participant on the analyzed patterns. No approach fully suited the envisioned collaborative pattern formulation approach to support project-wide knowledge management.

P1: Tedious Pattern Generation

Delayed knowledge sharing

Currently, most pattern collections are gathered over time by small groups of experts that have completely verified their findings before publishing them as design pattern collections or pattern languages, respectively. This "traditional" gathering process spans a longer period of time that can easily become equal to the project's lifetime until interested parties can benefit from the collections. For teaching and production purposes, this is certainly a good approach since the patterns are intended to provide knowledge that is ready to use. For the envisioned engineering and documentation process, however, workshop participants found it useful to document the whole way of idea finding, refining and validation process. As a consequence, knowledge sharing happens quite late in this process such that it is impossible to to benefit from the accumulated design knowledge during the development. Although the quality and validation work spent on compiling the patterns promise well-formulated and useful results, readers only have the possibility to consume the knowledge offered. They have to determine for themselves whether the pattern worked out for their problem situation.

P2: Closed Author Groups

Late sharing of available knowledge

The traditional pattern mining and structuring process in pattern collections or languages is mostly driven by a small group of authors. They share their knowledge and experience gathered during their specialization within a specific domain. Experiences from the workshop participants showed that this happens for many different research and application fields. The analysis of related work shows that groups of authors define sets of patterns from their long-term experience and structure them within an individual pattern language construction. Reiners et al. [2014b] call this procedure the "Pattern Guru Approach" in an intentionally provoking manner since a common way of collecting and formulating knowledge is rarely performed. Pattern ideas as promising sources of knowledge may be accessible very late during the engineering process.

P3: Lacking Influence on Pattern Formulation

No usage experience

Currently, there are only basic possibilities to feed experience back into existing pattern languages. The discussion on existing patterns is still hard to start and keep alive. Published patterns remain relatively static within documents. Though, to achieve improvement and progress, they need to be refined or discussed besides the actual publication in follow-up research, forums or conferences. In case pattern authors iterate over their patterns, they mostly integrate feedback from readers and practitioners. Still, in case of the printed publications, the reintegration has to be postponed to the next possible edition. Currently, there are only limited ways to extend and discuss research results.

Openness is only given in a passive way, i.e., the results can be read but not actively be extended or discussed. A defined pattern language may, at some point in time, loose its validity or can be refuted by practitioners who applied the pattern. New ideas and concepts cannot be included into the existing structure. Experiences made with the patterns cannot be reported easily or discussed with others. Proposals for refinements cannot be made. As a result supporting or refuting the pattern and therewith sharing experience with others is hardly possible.

P4: Extensibility and Actuality of Patterns

Pattern formulation beyond publication

Currently, patterns undergo a discussion process either together with domain experts directly or via writers' workshops supporting the understandability and applicability of a pattern. At some point in time, authors have to stop and present the derived pattern collection or language with the patterns derived so far. Without claiming that their languages are complete, they need to fix the derivation process for publication. The extensibility or later refinement is no longer given for the formulated patterns. The chance for further discussion of the published pattern was considered as high since the patterns are applied in practice, reviewed by readers and eventually reformulated and adapted for new pattern collections. That way, gathered knowledge can be clustered, structured, reused and combined. In order to cope with the dynamics during the project development, the knowledge base needs to be flexible in such a way that already integrated facts can be adapted or rearranged within the structure. New findings and approaches need to be included at the right place as well. Patterns have to be revalidated continuously over time.

P5: Lacking Reuse of Existing Knowledge

Integrate existing findings

Current pattern languages describe validated solutions to problems in specific domains. Especially for abstract patterns, knowledge that was predominantly formulated for a specific domain could be transferred to another one. The expert workshops revealed that different approaches with similar key concepts are hard to compare and reuse. This affects the recommendation of successful approaches and the availability of knowledge about bad designs. In system engineering, it can be supportive to reapply already gathered findings and respect them in system design. Especially when it comes to transferring knowledge from one domain to another, more specialized one with even more constraints and demands, it can be a hard task to organize and transform existing and gathered knowledge. Not only existing patterns are interesting to be analyzed. Lots of relevant knowledge is bound in publications, articles, internal studies, project deliverables or products. The important parts of knowledge have to be extracted from existing material. Regarding the analysis of related work and literature or product research, different denominations make it hard to identify and integrate existing knowledge.

P6: Lacking Transparency of Derivation

Relevance of derivation process

In printed publications, patterns are assembled after undergoing discussions in workshops or internal assessment processes within the authors' working environment. The patterns represented a reviewed and valuable source of knowledge shared by experts. Still, the derivation process itself in combination with the made assumptions and reasons for choosing a solution can be relevant for the reader of a pattern. Especially if authors integrate ratings into their pattern languages, the determination of a pattern's rating value is important to understand. Otherwise, and that was the impression by most of the workshop participants, the recommendations resemble more a rule of thumb instead a validation. Online portals allow readers to track the process of ratings that are supported by users. Inserted patterns are commented and discussed collaboratively. However, the derivation of the pattern itself is not shown but only a mature state that can be accepted or rejected by readers. Thus, the development process of a pattern together with its track of discussions were regarded as useful to decide whether the pattern itself is relevant for the treated problem. Feedback from pattern derivation workshops and recommendations based on voting were regarded as supportive for the readership.

P7: Long-Term Motivation and Inclusion of Contributors

Web-portals promising but active community needed

Workshop participants considered online discussion approaches as useful means to involve more people in the knowledge creation process but still see the danger that users loosing commitment to the pattern platforms since they are not directly integrated into the project's working environment. They still argued that online collections are valuable but need to include all relevant stakeholders of a project. In their view, mainly people who are directly interested in the pattern domain will contribute the most. Another identified obstacle was that possible participants may feel that their contributions will need a certain maturity until submission. As a consequence, knowledge may be held back instead of allowing for collaborative discussions.

P8: Lacking Recommendation and Guidance

Suggestions on reading order

Within a pattern collection or language, the entry points telling the readers where to start browsing the patterns and further guidance on a proposed reading order should be given in a pattern language. Most approaches provide navigating concepts for the pattern language but some lack from visualizations of the entire structure. Navigation then resembles following text links and the table of contents often needs to be consulted. Recommendations of proven combinations that are based on ratings and experience were furthermore regarded as helpful. In most approaches, suggestions on further reading refer to all related patterns. In case that decisions need to be made among alternatives no advice is given.

P9: Lacking Knowledge of Bad Practices

Only successful results are kept

Published patterns, often reveal information about working concepts that were successfully implemented. Only initial studies about a certain problem domain concretely outline deficits in order to justify and motivate intended research. During the workshops, ideas were formulated to include bad practices that are not trivial or were revealed unexpectedly during experiments and realization. Bad practices help avoiding traps and mistakes that early experiments and designs encountered. They document surprisingly failing approaches that turn out to be ineffective or counter-productive in practice Reiners [2011b]. Usually, unsuccessful implementations are rarely published or have to be distilled from results that mainly focus on successful aspects.

4.3 Requirements for an Evolving Pattern Library

Problem categories

The extracted problems from the workshops were assigned to four categories as shown in Figure 4.3. *Collaborative Pattern Formulation* deals with the derivation and formulation of patterns by every stakeholder of the project. Depending on the pattern's validity, it undergoes a *Continuous Maturation Process* that continuously improves the pattern formulation and ensures its usefulness. Since patterns are introduced and refined iteratively over time, a *Dynamic Pattern Library Structure* is needed that allows the introduction of new patterns, pattern relations and organizational elements. The provision of an overview of the pattern library's contents and structure as well as transparency and traceability of the maturation process is handled by the fourth category, i.e., *Management and Visualization* . It establishes the interface between the internal mechanisms of the pattern library and pattern authors as well as readers. In addition, pattern and structural consistency is handled by this component since it provides means to handle new content and structural change requests in an interactive and clearly arranged way.

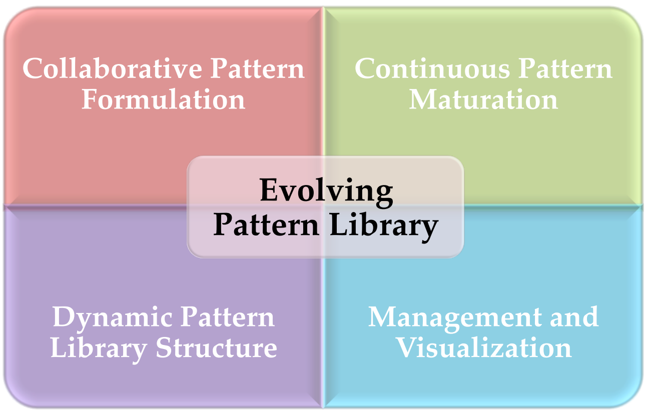


Figure 4.3: The four aspects treated by the evolving pattern library concept.

Initial requirements for first iteration

From the initial workshops in September 2011 a first set of requirements was formulated that tackled parts of the problems described in the previous section. Figure 4.4 illustrates the derivation of the requirements from the identified problems and aligns them with the appropriate components of the EPL. The requirements that were formulated for the first iteration of the BRIDGE Design Pattern Library prototype are described in Section 4.3.1.

Evaluation and further requirements

The prototype was evaluated later in January 2012 during a pattern collection workshop. Participants were asked to provide knowledge they already gathered so far and to formulate it as patterns by using the forms provided by the prototype. During that process new demands were gathered and elaborated in more detail during the second conception workshops in June and September 2012. Section 4.3.2 discusses the requirements gathered in these sessions that were supported by showing an online and interactive prototype.

4.3.1 Basic Requirements

Stakeholders and pattern maturation

The first formulated requirements concentrate on the involvement of many stakeholders in order to allow them to take influence on the pattern formulation process. The requirements focus on transparency of the concepts and means to reuse existing patterns from external collections. The development state of a pattern should also be reflected based on its applicability and validity.

R1: Availability of Knowledge

Tackles Problem(s): P1: [Tedious Pattern Generation], P2: [Closed Author Groups]

> *The assembly and refinement of knowledge should be encapsulated within patterns and always be available in any development state.*

Responsible Component(s): Collaborative Pattern Formulation, Continuous Pattern Maturation

Reflect iterative development

Description: According to the envisioned collaborative pattern formulation process, patterns should reflect the developments from different work packages within the research project. Concepts that are still under development are understood as ideas that still need validation. Over time, the formulation of patterns should become more complete and therefore show the progress and achievements of solutions to the tackled problems.

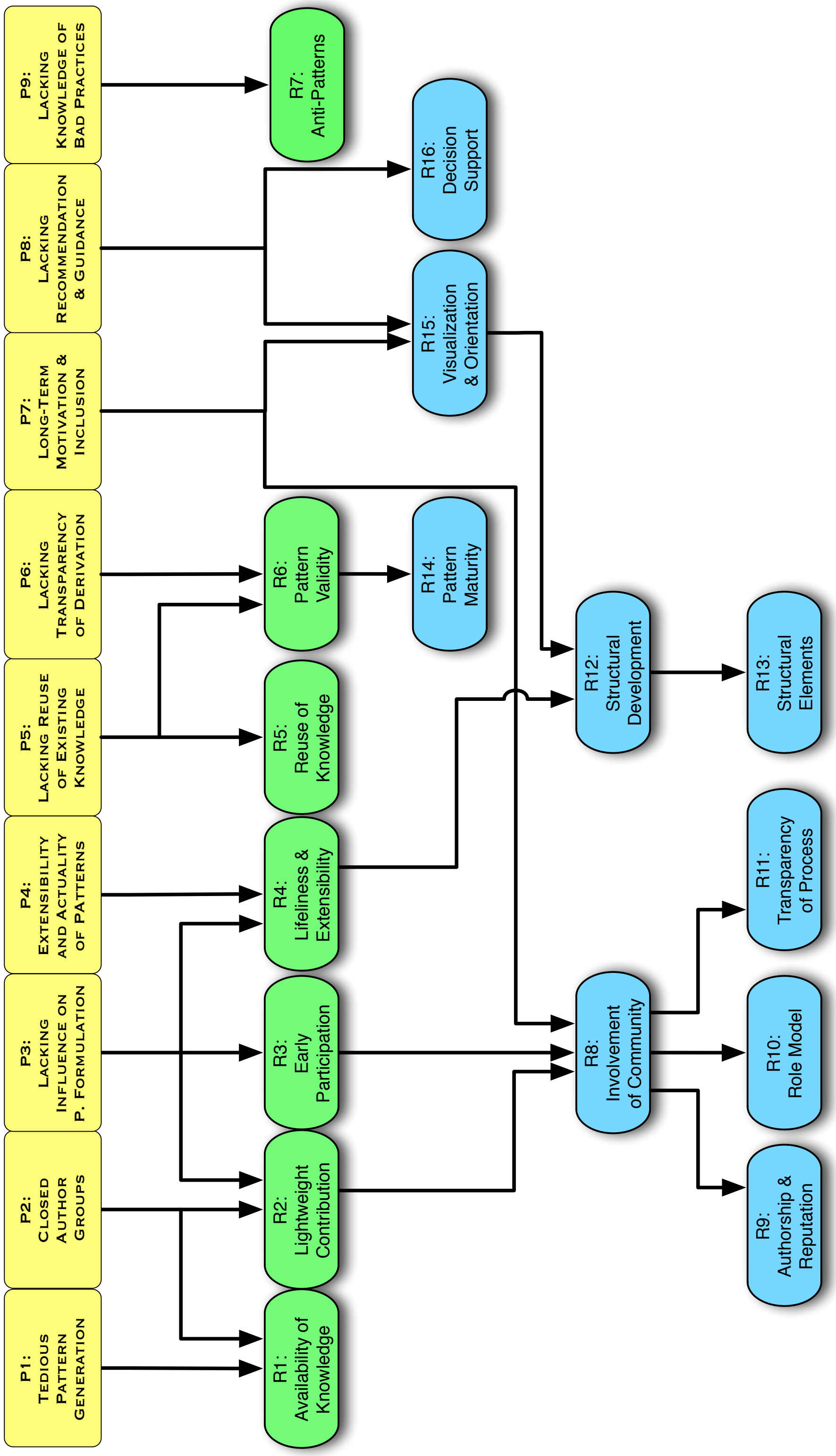


Figure 4.4: Gathered requirements based on the identified problems from the analysis of existing approaches.

R2: Lightweight Contribution

Tackles Problem(s): P2: [Closed Author Groups], P3: [Lacking Influence on Pattern Formulation]

> *Easy ways to access the pattern library must ensure that the time needed for contributions, i.e., pattern formulations as well as feedback and comments or new solutions, remains as short as possible.*

Responsible Component(s): Collaborative Pattern Formulation, Continuous Pattern Maturation

Quick contribution to knowledge repository

Description: An evolving pattern library needs to serve all project members as documentation and exchange platform but also serve as an understandable and easy-to-use project knowledge repository for new members. Duplicates of information must be avoided and different formats must remain consistent after partial updates and remain compatible for other work packages. Contributions should be made "on-the-fly" during the project work. Project members must be relieved from the load of lengthy documentation and thus encouraged to quickly promote their knowledge.

R3: Early Participation of Stakeholders

Tackles Problem(s): P3: [Lacking Influence on Pattern Formulation]

> *The approach should include all stakeholders during the whole project lifetime in the collaborative formulation and validation process.*

Responsible Component(s): Collaborative Pattern Formulation, Continuous Pattern Maturation

Include experience from all stakeholders

Description: Together with the lowered contribution threshold, an evolving pattern library should collect experience from all project participants at all stages of the development. Instead of leaving the experience management to a dedicated group of designers, and therewith potential design pattern authors, the envisioned approach involves *all* stakeholders from the beginning of a project in the knowledge creation, adaptation and management process.

R4: Liveliness and Extensibility

Tackles Problem(s): P3: [Lacking Influence on Pattern Formulation], P4: [Extensibility and Actuality of Patterns]

> *Whenever ideas or results are discovered, there should be means to directly formulate them as patterns. Discussions and feedback on propositions should be given directly and accessible for every project member, independently on time and location.*

Responsible Component(s): Collaborative Pattern Formulation, Continuous Pattern Maturation

Description: Patterns should not only be derived by authors and fixed with their publication. By providing a platform that supports the exchange among application designers, users and domain experts, the vividness of a pattern language may be increased. Community-based discussions on existing solutions therefore become possible that may lead to newly integrated aspects. Proven solutions can be re-evaluated based on experience gathered by others. This further influences the existing pattern library structure. This way, the pattern library is refined and extended continuously with the support of the participating community.

Community-based maturation

R5: Reuse of Knowledge

Tackles Problem(s): P5: [Lacking Reuse of Existing Knowledge]

> *The envisioned approach needs to introduce mechanisms to insert and adapt already proven designs to the context and them in the dynamic pattern library. Adaptation may be needed since the original pattern was eventually suited to special needs of a specific domain. This circumstance has to be regarded when formulating a pattern. A transformation into the pattern language's terminology be necessary.*

Responsible Component(s): Collaborative Pattern Formulation, Continuous Pattern Maturation

Description: Workshop participants stated from their experience that it is important to collect existing knowledge, share it among the members as early as possible and continuously validate gathered experiences during the whole project phase. Regarding the exploration of a new application domain, it can be important to adapt existing concepts to the problem at hand to avoid already known design flaws. This implies that patterns can be transferred across domains and be reused in different context, e.g., designing a widget on a mobile device and a desktop based system. It is very important that the pattern is marked as adapted since it may turn out that the concept is, against the initial expectation, not easily applicable in the new context. Consequently, the adapted design pattern has to be revalidated against the project scope.

Inclusion of existing concepts

R6: Pattern Validity

Tackles Problem(s): P5: [Lacking Reuse of Existing Knowledge], P6: [Lacking Transparency of Derivation]

> *Mechanism to provide the inclusion of evidence in favor of or against a formulated pattern should be integrated into the pattern derivation process. In addition, votes for or against successful applications of the pattern should be collected from the community.*

Responsible Component(s): Continuous Pattern Maturation

Iterative and parallel validation

Description: Proposed patterns and solutions must be validated over time. Within the examined approaches, mostly domain experts shared the pattern when they were certain about its usefulness. Additionally, in web-based portals, the community is able to vote for or against the pattern. For the envisioned process within a research project, validation will take place in parallel to the formulation of a pattern. Thus, *evidence* is found during the development and validation work that supports or refutes the proposed solution. Additionally, experience from practitioners should be used to show a pattern's usefulness. Both, evidence and experience influence a pattern's maturity and therefore its validity.

R7: Anti-Patterns

Tackles Problem(s): P9: [Lacking Knowledge of Bad Practices]

> *Non-trivial, surprisingly failing approaches should be documented as anti-patterns within the pattern language. Existing patterns may change to anti-patterns and vice versa, depending on the support or refutation of the formulated pattern by the evidence.*

Responsible Component(s): Collaborative Pattern Formulation, Continuous Pattern Maturation

Anti-patterns safe time

Description: Statements from the workshops showed that Anti-patterns were regarded to support the development of new concepts and ideas since approaches for best practices directly show what is good to be done. Approaches also stating bad examples could, on the other hand, avoid time-wasting and frustrating realization attempts as well as encourage to change parameters of an approach.

Patterns may change to anti-patterns

During the workshops, participants pointed out the possibility, that even promising patterns may turn out as anti-patterns over time due to technological, organizational or social changes. The concept of anti-patterns must be flexible enough such that existing patterns are constantly revalidated.

4.3.2 Advanced Requirements

Collaboration, development and visualization

The first BRIDGE Pattern Library prototype was realized based on the initial requirements discussed in the previous section. During a follow-up pattern collection workshop in January 2012 and two more workshops to further drive the conception of the collaborative pattern formulation process in June and September 2012, additional requirements were formulated. The next iteration of the EPL approach concentrated on better supporting the community-based pattern engineering process. Aspects on visualization as well as the development of the structure of the design pattern library over time became more important. Domain knowledge and validated approaches should be publicly accessible. Project members need to be encouraged to contribute their knowledge during all project phased and at all stages of maturity.

R8: Involvement of Community

Tackles Problem(s): P7: [Long-Term Motivation and Inclusion of Contributors]
Extends: R2: [Lightweight Contribution], R3: [Early Participation of Stakeholders]

> *An open, community-based approach needs to provide lightweight mechanisms to support pattern formulations. The community must be able to provide examples for pattern usages and offer feedback channels to discuss the formulated patterns.*

Responsible Component(s): Collaborative Pattern Formulation, Continuous Pattern Maturation, Management and Visualization

Many participants

Description: In connection to the problems of closed author groups and ways to influence the pattern formulation process, community-based feedback mechanisms that are included into the pattern formulation process were considered as very supportive. This way, the individual pattern's development but also the development of the whole pattern library structure are influenced by all participants. From the workshop participants, collaborative approaches were seen as very promising. Every user of the design pattern library needs to be able to comment on activities and contributions of others. Thus, formulated patterns can be discussed by many potential reviewers. Experts in the field can concentrate on the proposed solution whereas non-experts, for whom the patterns also should be understandable, can concentrate on aspects like readability and understandability of the solution.

R9: Authorship and Reputation

Extends: R8: [Involvement of Community]

> *The connection between individual ideas and contributions must be ensured. If many users contribute to a pattern's formulation, they need to be connected to their personal parts of intellectual property.*

Responsible Component(s): Collaborative Pattern Formulation, Continuous Pattern Maturation

Link activities and users

Description: Assuming that contributing takes place within an open-minded, progressive community with the common interest of improving the contents of an existing pattern library, the workshop participants stated concerns that the individual achievements could be invisible to authors. There need to be ways to show individual activities in terms of pattern contributions, comments, votes and recommendations. This influences the individual contributor's reputation that can be an important factor on the individual motivation. Quality assurance of contributions can be combined with incentives as discussed by Prause et al. [2010a].

R10: Role Model

Extends: R2: [Lightweight Contribution], R3: [Early Participation of Stakeholders]

> *A role model based on the suggestions from the workshops needs to be formulated that handles access to the pattern library as well as rights for organizing pattern contributions and for maintaining the pattern library's structure.*

Responsible Component(s): Collaborative Pattern Formulation, Dynamic Pattern Library Structure

Dedicated tasks

Description: Since every stakeholder in the project is asked to take part in the pattern formulation process, a differentiation of users by assigning roles based on their experience and tasks within the project is regarded as necessary.

R11: Transparency of Process

Extends: R8: [Involvement of Community]

> *The evolving design pattern library must provide means of reflecting the community's activities, new contributions as patterns, examples, solutions or evidence as well as newly added structural elements. Additionally, patterns that need to be reconsidered must be made prominent in order to trigger the community's attention again.*

Responsible Component(s): Management and Visualization

Reasons for pattern formulations and acceptance

Description: Since every idea about problems and findings should be reflected in the design pattern library (cf. R2), the activities by the members need to be reflected by the system. Activities from the community, such as comments, evidence and votes must be shown to reflect the pattern's acceptance and development. Reasons why patterns are regarded as valid and important must be clear to the readers. The same is regarded as important for existing patterns. A pattern library must provide ways to see where activity concentrates and improvements on pattern formulations or the library structure need to be made.

R12: Structural Development

Extends: R4: [Liveliness and Extensibility], R15: [Visualization and Orientation]

> *A flexible hierarchy concept must be developed that can be iteratively refined over time, similar to the single patterns. Structural changes like the introduction of new hierarchy levels as well as removal of existing hierarchies without affecting the pattern formulations must be realized.*

Responsible Component(s): Dynamic Pattern Library Structure

Structure over time

Description: Many pattern languages first present more abstract, conceptual patterns on the relevant context that readers first need to be informed about. Often, hierarchy levels are introduced that structure the pattern language, e.g., by means of spatial, temporal or detail-related aspects. During the pattern formulation process, the entire structure of the pattern library cannot be determined from the beginning and needs to be developed collaboratively over the project's lifetime.

R13: Structural Elements

Extends: R12: [Structural Development]

> *Structural elements for pattern associations are needed. In addition to the conventionally used non-exclusive* OR *semantics, relations that express* AND *as well as* XOR *semantics should be provided to better structure large pattern libraries.*

Responsible Component(s): Dynamic Pattern Library Structure

Subsets and alternatives

Description: Within large pattern libraries, mutually dependent or excluding concepts can be derived over time. It is also possible that only one alternative can be realized from an offered choice. The considered approaches carefully selected the patterns to be included in the pattern languages and mostly avoid explicit decisions. In related work, authors suggest several, non-exclusive design alternatives. Conflicting approaches are then discussed within dedicated sections within the patterns. For the envisioned approach the demand for more structural elements that provides a strong and more expressive organization structure was stated during the workshops. A demand for structural elements that affect all direct descendants of a pattern was formulated. If a pattern is split up into sub-patterns in order to follow the concept of micro-documentations, all subsequent patterns are relevant for the solution. Elements for deciding between alternatives within a pattern structure were seen as necessary since the research performed within a project may reveal contradictory results.

R14: Pattern Maturity

Extends: R6: [Pattern Validity]

> *A flexible rule system needs to defined to determine a pattern's maturity based on different kinds of evidence supporting or refuting the proposed solution. The weights of the different factors must be adjustable.*

Responsible Component(s): Continuous Pattern Maturation

Pattern development states

Description: Since the approach aims at including and reflecting all states of a pattern derivation process, the current state of a pattern needs to be determined. Starting from early ideas of a pattern with parts of solutions that eventually still lack validation support, accepted and proven patterns can be generated via the process. Workshop participants agreed that there is a hierarchy between the factors depending on level of abstractness of the formulated pattern.

R15: Visualization and Orientation

Tackles Problem(s): P7: [Long-Term Motivation and Inclusion of Contributors], P8: [Lacking Recommendation and Guidance]

> *Smart visualization methods are needed to support the users' browsing and contribution activities. The following aspects need to be considered:*
>
> - *The current state of a pattern.*
> - *The pattern language structure as a whole.*
> - *A pattern's position within the pattern library structure.*
> - *Relations between the patterns.*
> - *User activities.*
> - *Indicators for improving a pattern's formulation.*

Responsible Component(s): Management and Visualization

Visualization of the structure

Description: Large Pattern languages need to provide orientation for readers. Different publications provide more and less elaborate visualizations. Web-portals often make use of hyperlinks. Still, the workshops revealed that a graphical visualization of the structure of a pattern collection or language help users to keep the overview.

R16: Decision Support

Tackles Problem(s): P8: [Lacking Recommendation and Guidance]

> *Starting from a certain entry point within the pattern language structure, ranked guidance through a sequence of patterns should be provided via showing pattern combinations that were often chosen by the users of the pattern library.*

Responsible Component(s): Management and Visualization

Description: Decision support for picking certain patterns or pattern collections is given in different strengths. Gamma et al. [1994], for example, provide checklists from which readers can derive whether a pattern is suited for a specific situation. A similar way is chosen by Tidwell [2011]. Borchers [2001] as well as Schümmer and Lukosch [2007] follow the Alexandrian approach by rating each individual pattern depending on the personal evaluation (influenced by discussions from workshops and other rounds). The rating represents the degree of a pattern's generality as well as its successful application. In combination with a large pattern library and structural elements as described by R13, not only individual pattern ratings but also recommended pattern sequences were regarded as important navigation and selection support by the workshop participants.

Decision aids for alternatives

4.4 Conclusion

The aim of the approach presented in this thesis is to use the concept of design pattern for creating interdisciplinary micro-documentations that are exchanged and created among all stakeholders of a research project. Existing pattern collections and languages have already proven to support the aspect of interdisciplinary documentation. However, adaptions need to made in order to combine the concept with the iterative development character of a research project. Documentation is generated during the project work and extends or refines existing results. External influences from other projects and related research also need to be integrated. In order to suit the concept of design patterns to the research project's needs, a research agenda was set up that allowed for iteratively engineering the approach of collaboratively formulating design patterns.

Need for modification of design patterns concept

An exchange of gathered experience with the Enterprise and Software Architectures research group at Reutlingen University that is part of the SOA Innovation Lab consortium revealed that similar demands and problems concerning knowledge formulation and exchange between consortium members occurred. Patterns were also considered as a possible medium for formulating knowledge. In a cooperative workshop, requirements were derived that result from two different domains. Though different backgrounds, the general demands and problems that were collected during the workshops

Experience exchange with external research group were similar and could be integrated into one common set of requirements for the envisioned approach.

Comparison of existing approaches

The demands for a first prototype were collected and compared against existing approaches. Based on the comparison, missing features were identified and formulated as problems. From the identified problems, a first set of requirements was formulated that were realized in a prototypical implementation of an online design pattern library. Pattern collection workshops and feedback sessions revealed new requirements that were fed into a second iteration for the collaborative pattern formulation concept.

Basis for approach

The second set of advanced requirements form the basis for a more elaborated concept of the evolving pattern library approach and pattern maturation process that are subjects of the following chapter.

Chapter 5

The Evolving Pattern Library Concept

Since the presented approach aims at supporting collaboration and knowledge exchange within a distributed research project, the attention must be drawn to the stakeholders and their interrelations. The problem scope presented in Chapter 3 further describes the challenges encountered during this kind of projects. Human resources are clustered by the participating institutions or industry partners who represent individual interests in the project's development. On a second dimension, *tasks* are defined and structured in *work packages* that need to be managed. Collaboration as exchange of needs and ideas as well as mutual support must be strengthened. Staff and activities cannot be regarded separately since project partners usually take part in different work packages and the same employees may be involved in multiple tasks. Communication between employees from the same institution usually works well in case their offices are spatially collated. In contrary, the regular and appropriate information exchange between staff and work packages must be ensured.

Stakeholder relations are important

Patterns as communication medium need to cover many aspects of the gathered project knowledge that is continuously updated and extended during the project lifetime. Since every participating partner organization is specialized on different technical, methodological or domain-related topics, different states and prerequisites of knowledge must be respected. The first goal when starting a research project must be that the partners better know each other's specializations and knowledge. The different specializations that represent parts of the whole project knowledge must be made accessible for the whole project consortium. This means that, for example, technology-oriented partners need to explain the capabilities, requirements, limitations and chances of their contributions such that domain experts know what can be expected. The latter need to provide insight into the peculiarities of the regarded domain that are not necessarily clear to every participant from the beginning of the project. They need to explain how processes work and what kind of regulations, rules or laws exist that either support

Specialized stakeholders depend on available knowledge or hinder the application of novel technologies. Interaction and application designers equally depend on that kind of information. As specialists in their field, they do not necessarily possess specialized knowledge of the domain and its inherent peculiarities. On the other hand, they may expect more benefits from existing technologies or underestimate its benefits. Other stakeholders, especially controllers, managers and users need to be informed continuously about the state of provided and available information. Participants that handle exploitation tasks need to understand the current pool of knowledge.

Challenge for the presented approach

Keeping this in mind, the envisioned evolving pattern library approach represents a mixture of gathered experience, rules laws and findings that need to be explained to many different people with different views on the project and understandings of the envisioned concepts. The inter-disciplinary composition of the project must be regarded by the approach. The collaborative pattern formulation process supports the activities of all project participants concerning idea finding and refinement processes by providing recommendations for pattern sequences, decision nodes, the integration of anti-patterns and the support of continuous updates and changes via a dynamic pattern library.

A central repository for pattern management

A central repository that is available for all project members is the key infrastructure of handling micro-documentations formulated as patterns. All collaboration processes regarding new contributions, feedback, changes and updates are managed via the repository. Aiming at supporting the transparency of the ongoing activities, the pattern approach can help to explore new ideas based on considerations by changing or extending parts of a described concept. Made mistakes and the repeated derivation of unsuccessful concepts can be avoided during the conception phase for new approaches. Collaboration must ensure the sustainability of formulated patterns. In one way this is supported by formulating solutions without referring to concrete technical descriptions. Relevant reasons, processes and methods should be described to ensure the applicability of the pattern in different contexts and therefore making it reusable. In another way, pattern sustainability can be provided by the community that regularly reviews and updates made contributions. Patterns that need to be reconsidered or were not read since long can be made prominent by the repository to let them regain the community's attention.

Core functionality

According to the four core functionalities of the evolving pattern library (EPL) concept, Section 5.1 describes the collaborative pattern formulation approach that encourages the support of all project stakeholders. The approach forms the foundation for the pattern maturation process described in Section 5.2 that reflects the continuous documentation, update and refinement activities in a iterative engineering approach in the scope of distributed research projects. The dynamic derivation of a pattern library's structure addressing hierarchies, relations and structural aspects are treated in Section 5.3. Additionally, the necessity of tracking activities within the pattern library is discussed. The section closes with the proposal of *pattern sequences* that are intended to support novel readers and to provide application paths of proven pattern combinations. Section 5.4 handles the fourth functional group of the EPL concept concerning *Management and Visualization*. The discussion takes place on an abstract level by presenting general suggestions that were derived from further design workshops and that are proposed in related approaches that present implements of supporting tools. For the case study that is presented in the next chapter, an instantiation of the EPL concept is shown together with a complete visualization approach.

5.1 Collaborative Pattern Formulation

From ideas over concepts to patterns

In order to integrate as many findings as early as possible in the pattern collection, the presented approach lowers the threshold for contributing to the library. This allows contributors to provide early formulations that may range from considerations and incomplete pattern ideas up to acknowledged and proven solutions according to the original pattern concept. Each formulation represents a hypothesis that is valid until confuted. The patterns are created during the engineering process or may originate from other existing pattern collections. In the latter case, these patterns eventually need adaptation concerning their formulation and problem context. Some guidelines to extracting of experience into a pattern applying a convenient style to a pattern are given by Meszaros and Doble [1997]. Their "Pattern Language for Writing Patterns" is also available online[1]. In summary, patterns should be:

- *readable* such that quick parsing of the contents is possible.
- *tailored* to the target audience, i.e., in the context of this work, the project domain and work package tasks.
- *understandable* by the audience.
- *structured* such that the meaning of the different parts of a pattern is clear.
- *self-explanatory* by providing meaningful names and examples as well as using a common notation that is understood by every member of the audience.

Iterative, collaborative knowledge formulation

One important aim of the approach is the parallel documentation of findings while performing research and development. In addition, iterative progress for certain aspects must also be reflected by the approach. In order to ensure influence from all aspects of the projects, every stakeholder must be given the chance to contribute and understand the formulations. The contribution of experience from all project members is directly included into the library's concept. This way, many pieces of information can be contributed and analyzed by many project members and benefit from the individual stakeholder's specialization. It is no longer necessary to nominate a pattern formulation and assessment team that exclusively deals with the task of setting up and maintaining the pattern repository. In contrary, since every project member only has limited time slots

[1] http://hillside.net/index.php/a-pattern-language-for-pattern-writing

for documenting knowledge and findings as patterns, formulations can be reviewed, corrected and improved by all participants that spend a minimal amount of time and concentrate on specific patterns at a time.

Natural language and hierarchical categories

The patterns are formulated in natural language and therewith support common understandability, although this kind of formulation holds the danger of lacking exactness, especially in technical topics. By starting with early formulations that reflect the current progress of the task or state of existing knowledge, the single pattern is refined and updated over time. Clarifications and additional explanations, are provided by the feedback from the community that rates on the factors readability, understandability of the solution and relevance to the project scope. In order to structure the patterns, hierarchical categories are created that interlink the single patterns similar to the approaches proposed by Alexander [1977], Borchers [2001] and Schümmer and Lukosch [2007]. The hierarchies reflect different levels of abstractions starting from domain-related rules and laws and more specifically formulating processes, techniques and application concepts. Detailed descriptions of solutions are presented on the lower levels. Time, discussion and evaluation needs to show whether the formulated contribution is a successful pattern, a surprisingly failing approach leading to an anti-pattern or if the early formulation turns out as not suited to the pattern library and eventually needs reformulation.

Submissions undergo an initial quality check

The derived pattern maturation process that is schematically depicted in Figure 5.1. Each newly submitted pattern undergoes a first semi - automated quality check process meaning that the system first ensures that all required pattern fields are actually filled. The submission is then forwarded to a group of pattern reviewers who decide whether the pattern can be published in the library or if certain formulations need to be changed again by the author. This quality check avoids flooding of the pattern library with inappropriate content. After a submission has passed the quality check, it is published in the library and is therewith open for the community-based discussion within the community.

Collaborative feedback

From now on, every registered user of the system can provide feedback to the pattern, its formulation and support or refute the pattern statement by providing more evidence in favor of the pattern or against it. Evidence can originate from references from literature or realizations in products or applied processes that support or refute the pattern's validity. The community can make suggestions to reformulate parts of the pattern, split it into more patterns or merge it with already existing ones.

Maturity states

A pattern's maturity changes over time. From an initial pattern idea that can be submitted as an *open problem* for which a solution first needs to be found, it may develop from a pattern *under consideration* to the more mature states *pattern candidate* and *approved pattern* . These states reflect the usage and support of the pattern. Patterns in the state of an *open problem* further support the approach of using patterns as communication and documentation medium of project activities. Either later results in the research field of the authors lead to a solution that is added later or other members of the pattern library have ideas and experience on how to solve the stated, and therewith treated, problem. Section 5.2 explains the different states and conditions a pattern needs to fulfill in order to reach the next higher maturity state. Periodically, the quality criteria and currentness of a pattern are checked and decisions are taken on advancing to the next maturity state or remaining in the current one. In case that a pattern has not been regarded for a longer period of time, the librarian needs to take a decision on making the community aware of the pattern again or removing it from the library into the archive.

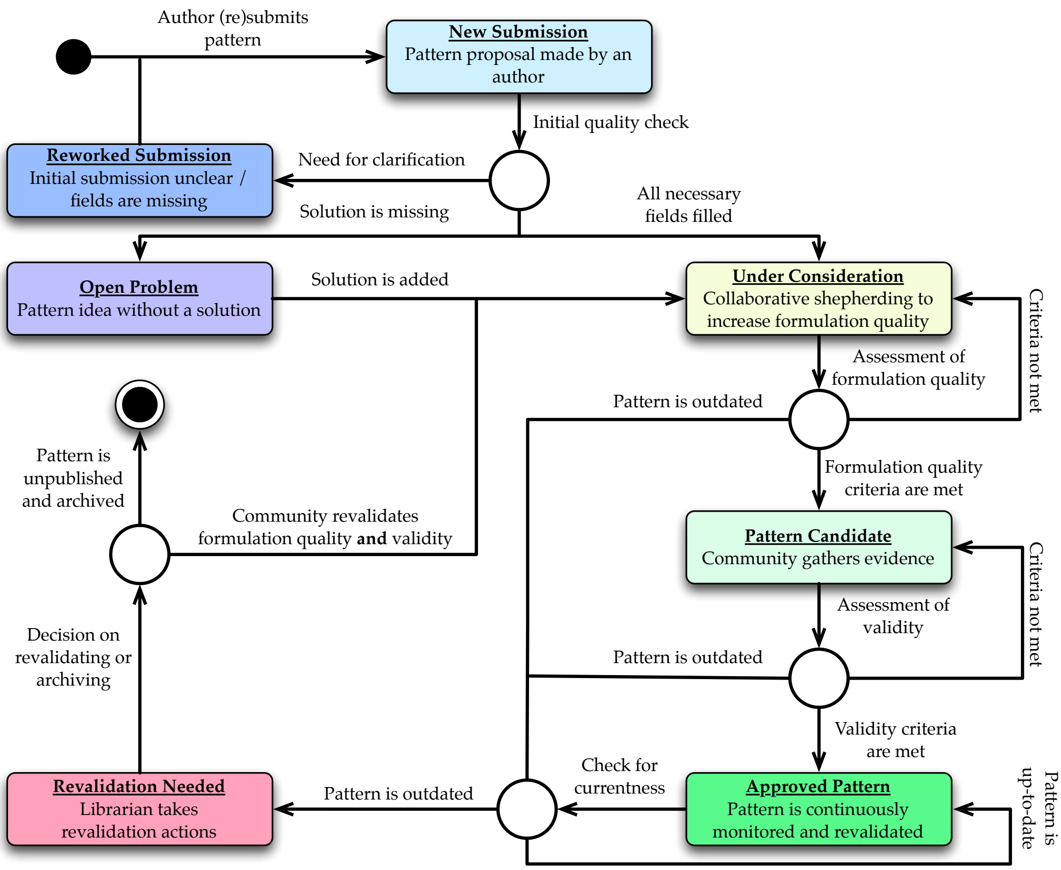


Figure 5.1: The different maturity states of the pattern maturation process.

The inclusion of anti-patterns as described in Reiners et al. [2011] is part of the evolving pattern library concept since surprisingly failing designs represent knowledge about flaws that were not obvious at design time and which should not be repeated. Anti-patterns support the development of new concepts and ideas since approaches for best practices directly show what is good to be done. Keeping track of bad examples can avoid time-wasting and frustrating realization attempts as well as encourage to change design parameters. Promising patterns or pattern adaptations from other collections may turn out as inappropriate over time. Revalidation may then change the pattern's state to a validated anti-pattern. Anti-patterns

Pattern structure and user roles

From this high-level view of the pattern maturation concept, the following sections describe the applied pattern structure and the derived role model for managing the contributions from the community and to establish collaboration as well as ensure the structural consistency of the dynamic pattern library.

5.1.1 Pattern Structure

Iterative completion of pattern fields

The used pattern structure is inspired by the approaches described by Alexander and others (cf. Alexander [1977], Borchers [2001], Schümmer and Lukosch [2007]) and provides a *flexible* set of fields that are filled over time during the maturation process, according to the lifecycle of information retrieval in a research project. The more mature a pattern becomes (cf. Section 5.2), the more fields need to be filled in order to improve its *completeness*. Based on the fields, i.e., the pattern's *formulation quality* and *validity* are determined and influenced by updated formulations, new validations and the community's feedback. Nevertheless, the fields that contain the pattern's *name*, *context* and *problem description* must be described by the author during submission for the first assessment of relevance. Each pattern should take into account the following principles:

- The patterns are formulated in natural language.
- The patterns must be easily understandable by non-experts.
- The patterns must be relevant for the project's domain.

Reading guidance

The recommended reading path of the pattern starts with the name that should already give an idea about the pattern's topic, followed by summaries of the problem and suggested solution. The contents should be easy and fast to read. Based on these fields, the reader is able to make a quick decision whether the pattern is suited to the current situation in the project work or information finding process. In this case the fields about the context, detailed problem description, solution summary and solution description are the next parts of interest. For the pattern as a whole, the fields are arranged in a different order as shown below since they are arranged in an argumentative way for reading the whole pattern. For quickly browsing the pattern and therefore capturing its essence, the relevant fields need to be shown in a prominent way. Alexander [1977] and Borchers [2001], e.g., use different kinds of type settings of the corresponding paragraph. For the reader's eye, these parts are easy to spot and focus on.

Pattern structure and fields

The fields *name*, *context* and *problem summary* are mandatory in order to pass the first quality check and trigger the pattern's maturation process (cf. Figure 5.1). The additional fields are intended to increase the pattern's aspects regarding its *completeness*, *understandability*, *maturity* and *validity*. The derived pattern structure that is used within the evolving library approach is as follows:

- The pattern's *Name* should be short and instructive, reflecting the solution to the problem being addressed. As in the traditional approaches, the name should be easy to remember and encapsulate the pattern's central statements such that it can serve the project's vocabulary.

- The *Hierarchy Level* is treated as a category. However, instead of simply clustering patterns, they shape the structure of the pattern library and therewith the relations between patterns, i.e. more abstract patterns are formulated in the upper hierarchies, more concrete patterns in the lower ones (cf. Section 5.3). The hierarchy level can be suggested by the pattern author but altered by community suggestions during the process. This way, the pattern can be moved to another hierarchy level depending on its abstractness of formulation.

- The *Pattern Maturity State* is determined by the rules of the pattern maturation process and depends on the pattern's *formulation quality*, with regard to readability, understandability and appropriateness, as well as its *validity* as explained in Section 5.2.

- Authors can initially mark a contribution as anti-pattern via an *Anti-Pattern Indicator*. In the interest of the amount of project knowledge that needs to be managed, only *non-trivial* flaws should be documented. Still, the anti-pattern indicator is continuously adjusted during the process that allows the contrary development of a pattern based on supporting and refuting evidence.

- The *Context* section relates patterns to each other according to Definition 2.2. In the context section of a pattern, the patterns that point towards the current one can be described such that only a brief summary of the context needs to be given. Further information can be found in the preceding patterns the context refers to. Often, a preceding pattern is extended by the current one that now tackles more specific aspects of a more general solution. A problem is examined in more detail and more specific solutions are described. With the help of the context, the reader is able to decide whether he possesses enough knowledge to understand the current situation the pattern describes or if he needs to read more preceding patterns in order to fully understand the current pattern's intention.

- The *Problem Summary* field briefly outlines the central problem the pattern tackles in order to allow the reader to quickly decide whether the pattern matches to the problem situation he is currently dealing with.

- The field on *Problem Details and Forces* further describes the problem context and discusses reasons that lead to the problem. Reasons can originate from external influences such as legal or technical restrictions that are further elaborated. These "forces" influence the proposed solution. When applying the pattern, the reasons for the forces as well as their impact on the solution need to be understood by the reader. Since the proposed solution may be well-suited for specific aspects but implies disadvantages on other aspects, different patterns on the same level may propose different alternative solutions. In the scope of exploring the project and development knowledge, the understanding of forces within the detailed problem discussion may lead to alternative solutions that have not yet been analyzed.

- The *Solution Summary* provides the central statement of how to solve the problem in the given context. The concise formulation of the solution serves the reader's decision making process whether the pattern is suited to his current situation but also helps to better remember the advice given by the pattern.
- The section on *Solution Details and Consequences* elaborates on the factors and reasons that lead to the solution. Explanations and considerations on the pros and cons with regard to the forces are given. This part should point out benefits but also discuss disadvantages that occur when following the advice. Here, the consultative character of a pattern as solution approach but also as knowledge source is emphasized. The solution is not to be accepted as a statement out of question but as consideration and elaboration of known possibilities and consequences.
- *Illustrations* enrich the explanation of the solution and further support the usage of the pattern as vocabulary. The pattern's name and central solution statement can be connected to the illustration therefore making it easier to remember and recognize the pattern again during browsing the pattern language. In the presented approach, multiple illustrations may be submitted as drawings, diagrams, photographies or videos. The pattern author is encouraged to primarily provide a key illustration in an image-based format to support the described mental connections.
- The mentioning of a pattern's *Pattern Origin* encourages the reuse of already existing and validated results from other projects or repositories that are relevant for the current project knowledge. The origin field differentiates between newly derived and introduced patterns. It may be necessary to adapt the latter kind of patterns since they may be formulated in a different, more specific or more abstract context but the essence of the pattern is relevant for the project and can be extracted. This circumstance has to be regarded when formulating the pattern. The insertion may need to adapt naming conventions and formulations and transform them into the pattern language's terminology. The approach distinguishes between three different categories:
  - *Derived from project*: The pattern was derived directly from the work within the project. Continuous formulation and validation need to ensure the pattern's validity.
  - *Adapted to project*: The pattern originates from external sources, but has been adapted to the project's context. Still, the need for validation is given. Preliminary project-external work was already put into the pattern's formulation and is used as evidence supporting the pattern as described below.
  - *Project-external*: The pattern exists in other related pattern collections. The pattern can directly be used in the current project scope since the project domain and pattern origin are closely related.

Regarding the latter two categories, all externally existing patterns must be referenced by their source and interests in intellectual property must be regarded.

- For each pattern, *evidence* is collected by the author or as feedback from the community that supports or refutes the pattern solution. All patterns, regardless of their origin, must be justified. Documented proven solutions, realizations as concept-cases, prototypes as well as published studies or reports represent evidence of difference importance. A pattern's evidence is essential for its validation that reflects the successful application of a pattern or avoidance in case of an anti-pattern, respectively. Domain rules, processes and legal regulations, for example, could be submitted as documents or links to official sites where they are maintained. For domain knowledge, video reports or interview recordings and transcripts could be the appropriate medium. Empirical, statistical or technical evidence may be given as links, pictures or publications. Section 5.2 analyzes possible weights of the different kinds of evidence depending on the pattern's abstraction level.
- At the end, references to *Related Patterns* detailing the current context are given. Alternatives and other patterns that are supportive are named in this section. The approach presented in this thesis foresees advanced semantics of relations that lead to partial and complete combinations and mutually excluding alternatives of succeeding patterns as described in Section 5.3.

Organizing the community

The process involves a large group of potential contributors to the pattern library, be it authors, reviewers or practitioners who apply and therewith validate the formulated patterns. Still, the access to the pattern library must remain effortless. Browsing and contributing must be possible without high stakes. On the other side, activities must be coordinated and checks need to be performed in order to ensure the patterns' quality as well as to maintain the pattern library's structure. Additionally, the feedback as comments, suggestions for improvements and demands for pattern placement within the pattern language hierarchy need to be supervised and finally granted. The role model presented in the next section tackles these challenges by assigning duties to special users and boards.

5.1.2 Roles and Advisory Boards

Differentiations

The approach introduces a light-weight role model that allows to directly access the pattern library's contents but distinguishes between different activities concerning the contribution of new patterns, evidence, feedback and management tasks that concern the publication of contents and structural changes. Users can act in multiple roles, depending on the task they currently perform. Taking democratic decisions is proposed by the concept to ensure quality of the contents without the need to involve every user of the pattern library in every decision. For this reason, two advisory boards are established for different kinds of decisions. The member size of the advisory boards can be determined specifically for the project size. In the case that the advisory boards only consists of one member, the member's role is similar to an administrator and quality assurance manager. Members in the board should not be responsible for deciding on their own submissions since a peer-review approach should be kept. The approach keeps the number of needed votes for a decision in each board acceptance is configurable depending on the project aims and the library size. Figure 5.2 shows the five different roles and two advisory boards.

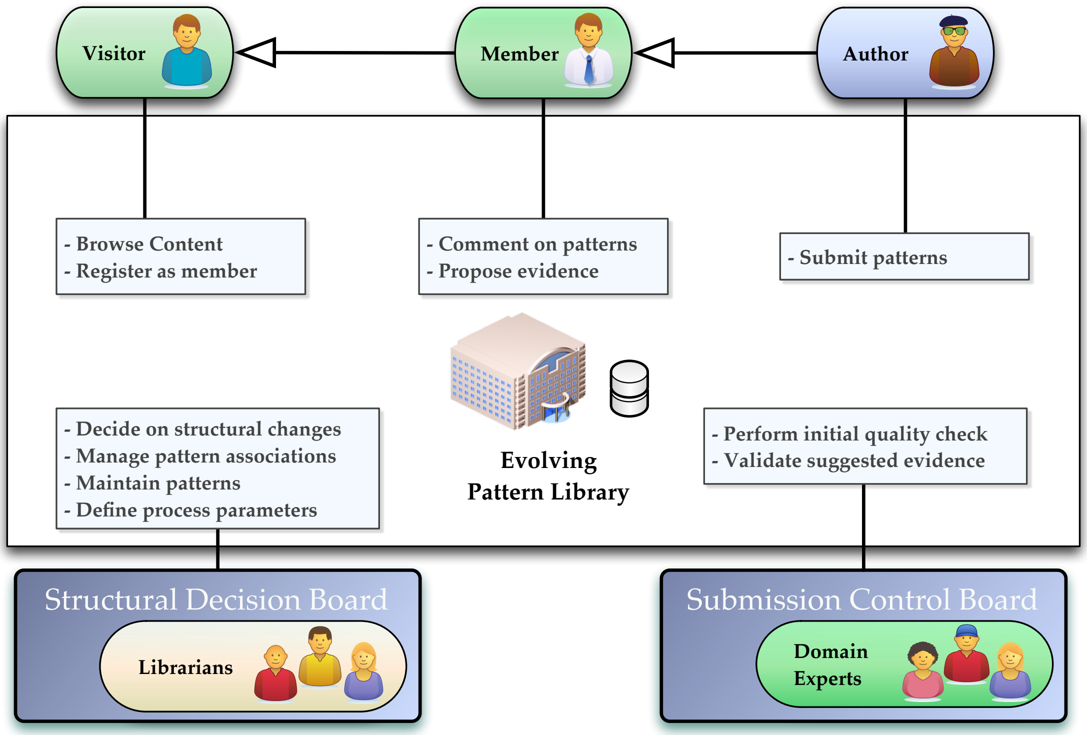


Figure 5.2: Overview of the defined roles of the pattern maturation process. *Librarians* and *Domain Experts* as board members take decisions on the library's content and structure[2].

Registered contributors

Except for visitors who are treated like guests that are allowed to read all contents of the library, all other roles need to create a user account. User identification is needed for tracking changes in the library history (cf. Section 5.3.2) within the library and communicating with the user concerning clarification or inquiries for further supporting the process. Additionally, the registration intendeds to stronger include interested users into the formulation and maturation process and be able to notify them about changes. The registration process should only ask for a user name, a valid email address and a password. Further information about the user's profile is completely optional. Except

[2]Icons taken from the UX People Stencil for Omnigraffle with kind authorization by Peter Morville and Jeff Callender. The stencil is available at https://www.graffletopia.com/stencils/639.
Library icon taken from the "Standard City" icon sets by Aha-Soft at http://www.iconarchive.com.

for the *Librarian*, all roles are extending the possible activities. This way, members can act as contributors and validate content submitted by others.

The approach includes the following roles: Derived roles

- *Visitor*: Every visitor can browse the design patterns and read a pattern's details. Since the approach wants to track changes and relate them to specific users in case that there is a need for clarification, other activities in the process ask for becoming a member.
- *Member*: Registered users are allowed to make comments on existing patterns and the library structure. Further evidence that supports or confutes the pattern can be proposed and needs to be accepted by the *Submission Control Board* that consists of *Domain Experts*. This way, a possible suppression of refuting evidence can be avoided by letting a steering committee check and eventually include evidence.
- *Author*: This is an extended role which is automatically assigned to members who submit a new pattern . The submission is linked to the author who is responsible for taking care of maintaining the pattern inside the library. Thus, *Members* are only assigned the *Author* role for their own patterns. Whenever feedback or change requests are subject of discussion, the pattern responsible is always included. This mechanism is applied to make sure that the original intention of the pattern authors is kept. In case the original author is not available at the time of a change request, the *Librarian* assigns tasks for reformulation to other pattern authors who take over the responsibility for the pattern's development.
- *Domain Expert*: Users in this role are responsible for the continuous quality assurance of the formulated patterns in the library. Newly submitted patterns are first checked by representatives of this role in the *Submission Control Board*. As the initial quality gate (cf. Figure 5.1), they assure that the submission fulfills the minimal requirements with regard to a name, context description and a problem summary as well as a solution summary. *Domain Experts* also validate and accept suggested evidence for or against a pattern and are responsible for checking the references. In case the pattern author disagrees with suggested evidence the discussion is taken up with the *Submission Control Board* members.
- *Librarian*: As member of the *Structural Decision Board*, this role decides on the acceptance of suggested changes to pattern relations, i.e., adding, modifying or removing relations. In case for a strong demand for introducing or merging hierarchy levels within the library, this decision is also made by the committee. *Librarians* maintain existing patterns or *open problems*, respectively and remove outdated or non-developing patterns. The rules of the whole process are adjustable with regard to weights for a pattern's maturity, the member size of each committee and the necessary acceptance rate as well as the configuration of the applied pattern hierarchy structure. Therefore, the librarian acts as system administrator who has access to all the data of the library and parameters of the applied rules.

Role assignment

The only automated role assignment takes place when *Visitors* register at the library and become *Members*. The latter are related as *Authors* to their own submitted patterns. The other roles are determined by the project management when setting up the pattern maturation process. Later, *Librarians* manually assign roles and memberships to advisory boards throughout the project's development.

Towards a pattern maturation process

After setting the foundation of the pattern library by defining the overall formulation process, the used pattern structure and the role model that is needed to coordinate the users' activities as well as the internal mechanisms of a pattern's development are described in the next section.

5.2 Pattern Maturation

Formulation quality and validity

The overall aim of the pattern maturity concept is to improve a pattern's formulation quality and validity such that it can be reused in different contexts and represents relevant and competent project-related knowledge. During the workshops held, the quality gate of this state was inspired by the quality criteria defined by Borchers [2001], who derives a pattern's formulation quality, among other factors, from its readability for different domains and coverage of the design dimensions. Another source for formulation quality is given by Khazanchi et al. [2008] who consider practical factors like feasibility, predictability and plausibility. Wurhofer and Obrist [2010] present a quality criteria framework for patterns that analyzes a pattern's findability, understandability, helpfulness, empirical verification and overall acceptability. Especially the aspects on understandability and helpfulness are assessed in more detail by the derived criteria framework which is based on related approaches. The guidelines for formulating patterns by Meszaros and Doble [1997] provide further inspiration on the extracted aspects that are more detailed within the single patterns.

Quality criteria

In order to keep the process and pattern assessment light-weight, three quality criteria for the pattern assessment were regarded as relevant during the workshops:

- *Readability:* The pattern must be written in a fluent and quickly to digest way and the structure and length of each pattern field is adequate. Illustrations and examples must fit to the description and presented solution. Lengthy or ambiguous parts must be avoided and reformulated.
- *Understandability*: The used vocabulary must be free from expert jargon and thus be comprehensible by every member. The descriptions must be complete and straight to the point without distractions from the pattern's original topic.
- *Appropriateness*: The pattern must fit into the project context such that the tackled problem and described solution are relevant to the project domain. It must be accepted by the participants and represent a validated source of knowledge the project benefits from.

A pattern's maturity is continuously determined during the maturation process and based on user feedback as comments on its formulation quality as well as its validity. The latter is supported by the provision of evidence in favor of or against the pattern. Applied rules and processes for increasing a pattern's maturity are described in the following section, according to Figure 5.1.

Continuous assessment

5.2.1 Formulation Quality and Patterns "Under Consideration"

A newly submitted pattern is first examined by the *Submission Control Board* that checks whether the necessary fields, i.e., *Name*, *Context*, *Problem Summary* and *Solution Summary* are filled. In case the solution but the other required fields are provided, the *Submission Control Board* publishes the submission as an *Open Problem*. This way, the problem description in a specific context get accessible for all project members. In case that a hierarchy level is proposed by the pattern author, the board checks whether it fits in the current hierarchy of the pattern library or makes adjustments. The pattern resides in this states until a solution is added by the author.

Submission as pattern idea or open problem

As soon as a solution is suggested, either directly at the time of submission or added later, the pattern reaches the state *under consideration*. The primary focus in this state lies on the formulation quality of the pattern. Therefore, the community-based collaborative shepherding process on the patterns' formulation starts. All members of the pattern library are encouraged to comment on the pattern formulation and give advice and suggestions for improvements or different solutions. Pattern authors decide on the integration of the given feedback.

Collaborative formulation review

A minimum amount of users rates on a set of formulation quality aspects. Throughout the workshops, a Likert scaling based rating mechanism on the aspects concerning the pattern formulation quality was considered as useful for the community. The quality gate for ensuring the formulation quality of a pattern is generically defined as follows:

Generic rating aspects

DEFINITION 5.1: FORMULATION QUALITY GATE
A pattern's formulation quality gate is determined by aspects a_i that each needs to be fulfilled to a level

$$l_i \geq r(a_i, u_{min})$$

$$\text{where } 1 \leq i \leq n \text{ and } n, u_{min} \in \{\mathbb{N}^+\}.$$

The rating function $r(a_i, u_{min})$ is defined within the individual project's scope.

The formulation quality factors and the parameters for the quality gate's threshold are maintained by the *Librarian*. Over time, a pattern formulation may turn out as an inappropriate or a non-supportive formulation. The commenting and rating phase intends to uncover such cases as early as possible. In parallel, the collection of evidence may take place but is not yet considered in the current maturity state.

Librarian supervises parameters

5.2.2 The Validity of a "Pattern Candidate"

Indicators for a pattern's validity

As soon as the formulation quality of the pattern is ensured by the described quality gate, the submission is in the state of a *pattern candidate*. The next aim within the process is to ensure the *validity* of the pattern. This is achieved by supporting or refuting the pattern by assigning evidence accordingly. For the determination of a pattern's validity, an indicator denoted as *Pattern Maturity Indicator (PMI)* is introduced that extends the considerations presented by Grill and Blauhut [2008] who make use of the number of indicated successful applications to determine a pattern's validity. The presented approach takes into account more indicators that, in addition, can be taken as support or objection for a pattern, i.e., increasing or decreasing its validity. The different factors should cover evaluations, theoretical considerations, existing circumstances as well as rules, realizations and showcases. For each evidence factor, the sum of all assigned cues is calculated. A positive values for c indicates support for a pattern, a negative one stands for its objection. Thus the *Evidence Factor Calculation* used for the approach is defined as:

DEFINITION 5.2: EVIDENCE FACTOR CALCULATION
Each evidence factor e for a pattern p is defined as the sum of assigned cues c:

$$e_p = \sum_{i=1}^{n} c_{i_p}$$
where $\{c_i \in \{-1; 1\}$.

Types and weights

Depending on the project's properties, an arbitrary number of *evidence factors* with corresponding *weights of importance* can be defined. According to the project circumstances, the relation between different aspects can be adjusted. The general definition of the *Pattern Maturity Indicator* (PMI) is given as follows:

DEFINITION 5.3: PATTERN MATURITY INDICATOR
The *Pattern Maturity Indicator PMI* for a pattern p is determined based on given evidence factors e_i assigned to p with a relative importance weight w_i as

$$PMI(p) = \sum_{i=1}^{n} w_i e_{i_p}$$

where $\sum_{i=1}^{n} w_i = 1$, $\{w_i \in \mathbb{R} \mid 0 \leq w_i \leq 1\}$.

From the user workshops that served as preparation for the development of a pattern library prototype, the following evidence indicator were extracted, serving as example at this point:

- Scientific publications evaluating approaches that make use of the pattern.
- Successful applications of the pattern indicated by members.

- Prototypes implementing the solution proposed by a pattern.
- User studies that empirically or qualitatively evaluate the concept described by the pattern within a product, process or prototype.
- Interviews as well as video or audio recordings providing empirical and qualitative data.
- Validations that empirically or qualitatively evaluate the concept described by the pattern within a product, process or prototype.
- Commercial realizations in which the pattern can be recognized or that explicitly use of the pattern.

Different PMI definitions per hierarchy

In combination with the dynamic hierarchy concept as described in Section 5.3, the PMI can be defined per hierarchy level. Since the approach aims at providing project-related knowledge, the hierarchy also needs to provide levels that are concerned with general rules, laws or processes that need to be taken into account for the whole project scope. Other levels may concentrate on processes, human- or machine-related aspects or, even more concretely, concepts or implementations. Evidence for the patterns then may be of different kinds and importance. Thus, the approach foresees the definition of *hierarchy-dependent pattern maturity indicators* defined as follows:

DEFINITION 5.4: HIERARCHY LEVEL DEPENDENT PATTERN MATURITY INDICATOR
For each hierarchy level l in a pattern library P, a specific pattern maturity indicator PMI_l can be defined as

$$PMI_l(p) = \sum_{i=1}^{n} w_{i_l} e_{i_p}$$

$$\text{where } \sum_{i=1}^{n} w_{i_l} = 1, w_{i_l} \in \mathbb{R} \mid 0 \leq w_{i_l} \leq 1\}.$$

Validation quality gate

Evidence is collected continuously over time. Validations during the project work together with ongoing application of the pattern may indicate its development into an anti-pattern and vice versa. The presented process allows the flexible provision of evidence such that the anti-pattern indicator may change gradually. Together with Definition 5.4, the *validation quality gate* for each level in a pattern hierarchy is defined that a pattern needs to pass in order to reach the final maturity state:

DEFINITION 5.5: VALIDATION QUALITY GATE
The *validation quality gate* t_l for a pattern p in a hierarchy level l is defined as:

$$PMI_l(p) \geq t_l \text{ where } t_l \in \mathbb{R}.$$

5.2.3 Pattern Approval

Approved patterns remain under supervision

After a pattern is validated by the provision of sufficient evidence for its maturity indicator as defined by the validation quality gate of the corresponding hierarchy, it reaches the state of an *approved pattern* meaning that it can be considered as reliable source of project-related knowledge and problem-solutions within the project domain. The validated pattern reflects the project work and findings that were gathered continuously. The development history of each pattern shows the iterative steps it may have undergone until reaching this state. Still, the process does not end but continuously checks for new comments and evidence provided by the community. User inactivity with regard to reading a pattern or providing feedback for a long period, triggers a message for the *Librarian*. He then decides on asking the community for revalidating the pattern. In case the pattern is regarded as not valid anymore, it is unpublished but remains in the library's archive.

5.3 Dynamic Pattern Library Structure

Knowledge categorization

The presented approach aims at formulating patterns for many different kinds of project-related knowledge such as domain- and process-related backgrounds, guidelines and applied practices as well as technical realizations together with their iterative advancements and validations. This means that knowledge is encapsulated in rules, laws, guidelines, processes, concepts and technology that is assembled by many different contributors and specialists in the project's work packages and external advisory boards. Bringing them all together in one pattern library is a challenge. As Figure 5.3 suggests, the approach proposes a pattern hierarchy starting from abstract information that is valid throughout all further pieces of knowledge. At the upper levels describe domain-related conditions and restrictions. They provide the foundation for the design space of concept cases, realizations and project assets on the lower levels. The lowest, most concrete realization level leaves the standard pattern convention of technology-independent descriptions but is essential for documenting project results. More conceptual results are explained in patterns of preceding levels. Still, the concrete descriptions are intended to give inspiration for design alternatives that can again be formulated as patterns. The hierarchy levels serve on the one hand as pattern categories, on the other hand they allow to gradually structure the knowledge such that the next lower hierarchy level of a category concretizes the current one.

Assignment of patterns to existing or new hierarchy levels

When proposing a new pattern to the pattern library, the author can suggest a hierarchy level or let the *Submission Advisory Board* decide. The decision does not have to be final. It may turn out over time that the collaborative shepherding activities suggest to change the pattern's hierarchy level or to introduce a new hierarchy level. The *Librarian* then adds the new hierarchy level and relocates the pattern within the library. Relations to and from other patterns need to be checked for validity for this step. Depending on

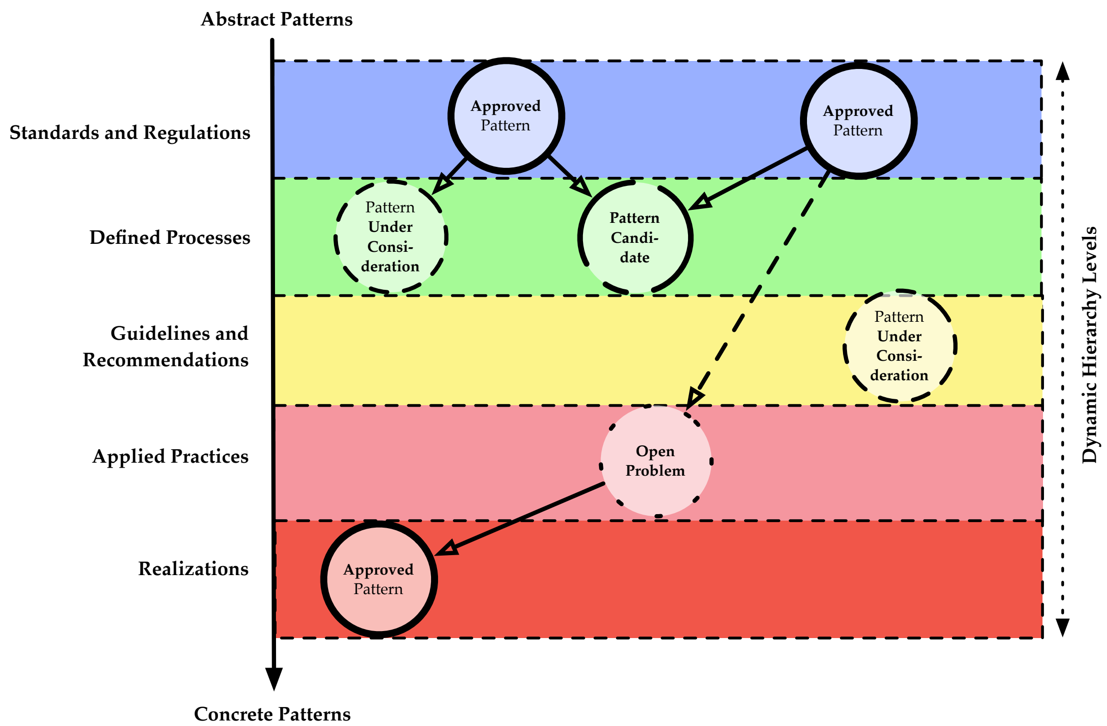


Figure 5.3: Suggested hierarchy levels from abstract to more concrete patterns.

the community's feedback, the library is subject to continuous change and therewith remains flexible.

Pattern relations develop over time

Patterns in all introduced maturity states are published in the library after they passed the initial quality check and are only removed in case they turn out as inappropriate or outdated. At submission time, it is not always possible to associated patterns to others. The early state of the pattern library only provides a repository for submitting knowledge as individual patterns. Relations are added over time and interconnect patterns. This may imply that smaller parts of patterns need to be reformulated and adapted in order to fit into the context, i.e., the incoming edges as well as the references to other patterns represented by outgoing edges. Figure 5.3 also shows examples of patterns in different maturity states that are only partially connected illustrating that missing associations can be suggested at a later point in time. Connections that range over more than one hierarchy level may suggest that a pattern is missing at that point. A graph visualization similar to the depicted one is encouraged to show the current structure of the whole pattern library.

Board decides on change requests

New pattern relations or changes to existing ones are proposed by the collaborative shepherding process as comments or direct demands that need approval by the *Structural Decision Board*. Proposed structural changes are only visible to the board and *Librarians* who perform approved changes. The demands for changes may result in duplicate, inconsistent or transitive relations that need to be removed. Alternatively, patterns that depend on each other may be identified and then associated with each other. A change or deletion of relations, respectively, undergoes the same proposal process as newly added relations. In Figure 5.3, the propositions of relations are illustrated as dashed arrows whereas accepted connections are shown as solid relations. The strength of the enclosing circles represents a pattern's maturity state. Some connections are not yet made to show that it is thoroughly possible that patterns are not yet or only partially interconnected. The dynamic structure is intended to reflect the acquisition of project-related knowledge over time which is gathered from domain analysis, prototyping and validation.

Flexible hierarchy model

The approach does not exactly prescribe the hierarchies as named in Figure 5.3 but furthermore encourages contributors to think about where in the structure the submitted pattern resides best or whether an already positioned pattern should be moved to another hierarchy level. Additionally, it is possible for contributors to make proposals for new hierarchy levels that extend the structure and allow for refining existing pattern categories. Alternatively, in case of a demand for more vague categories, hierarchy levels can be merged or removed from the pattern library. The presented hierarchy represents initial suggestions based on the results of the user workshops and interviews conducted so far. The only demand for the hierarchy is that it must be concretized towards the lower levels.

Introducing AND semantics

Conventionally, relations between patterns express OR semantics. Related patterns tackle a current problem in more detail or discuss alternatives. The presented approach conceptually introduces two additional types of relations. AND semantics were demanded during the user workshop sessions that allow the inclusion of n following patterns in case they are all relevant for further understanding the problem. The follow-up patterns could concentrate on different aspects of the problem and are therefore formulated separately.

Introducing XOR semantics

XOR semantics were regarded as necessary as soon as the pattern library exceeds a size limit at which patterns represent alternatives and therefore exclude each other. At some point in time, decisions may be needed on the further development of concepts and designs. The same situation may occur at a higher level in the hierarchy as soon as provided knowledge is specialized to parts of the domain. Hierarchies alone cannot solve this problem. Therefore, the evolving pattern library concept introduces XOR semantics for relations serving as *decision points*.

5.3.1 Pattern Aggregation and Decomposition

Need for structural refactoring

Over time the pattern library evolves and starts growing, different refactoring measures may become necessary concerning the management and change of pattern formulations and relations between them. The feedback of the community plays an important role in this community-based formulation and contribution process besides the monitoring tasks of the *Structural Decision Board*.

Split up complexity

In case that the formulation of the problem and solution sections within a pattern have become to all-embracing, it gets hard for the reader to clearly structure the single pieces of information. The approach of using patterns as micro-documentation interdicts this situation. Figure 5.4 shows such a situation. The *Structural Decision Board* needs to extract large parts within a pattern and split them up into separate patterns. Eventually, a common preceding pattern needs to be defined if there are still commonalities shared by the new patterns that suffice for the formulation of a predecessor. The relations need to be adapted to the new situation by either keeping incoming context edges with the newly formulated predecessor or connecting the new pattern group directly to patterns of a higher hierarchy. In case that no predecessor can be determined, the pattern remains isolated from the higher levels until a valid relation can be found or formulated. The same situation may occur that many solutions for *open problems* were formulated that cannot be merged or no best solution can be determined. As a result, the different solutions need to be reformulated as more detailed successors of the original one and connect to the more abstract problem and solution formulation via relations with XOR semantics.

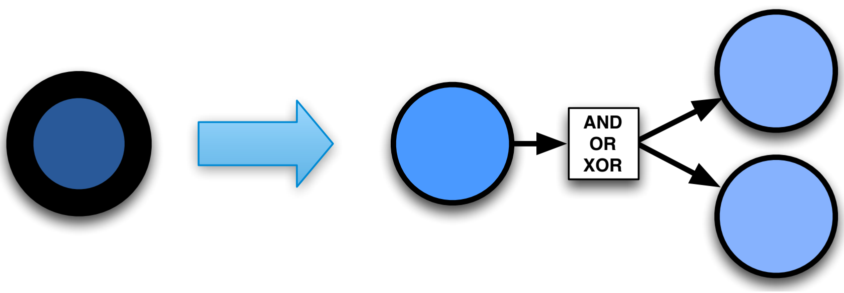


Figure 5.4: A pattern containing too much information or alternatives for solution is split up into succeeding pattern with specializations.

New relation semantics for refactoring

The type of the relation is also case-depending. OR semantics are appropriate in case that some or all of the newly formulated patterns should be considered. The enforcement of considering all formulated pattern via AND semantics should be chosen in case that the splitting only happened due to structural and readability reasons but the patterns need to be taken into account completely. A decision, i.e., XOR semantics may become necessary in case that the contents fragmented and different approaches are merged into one pattern. In fact, such a strict separation should be avoided in advance by the *Structural Advisory Board*.

Merge and remove duplicates

The opposite situation is described in 5.5 in which similar information got formulated at different points and therefore scattered among different patterns. In this case, the similarities need to be merged in one pattern and eventually destroy the others. The commonalities are extracted and formulated as a more abstract pattern to which the similar patterns are connected as children. A very detailed pattern may be split up into one pattern concentrating on the core problem and several sub-patterns each dealing with different aspects in more detail. Alternatively, a common predecessor can be created in order to merge the similarities. In the resulting situation, the patterns are used as differentiations with the same possibilities for the relation semantics as described.

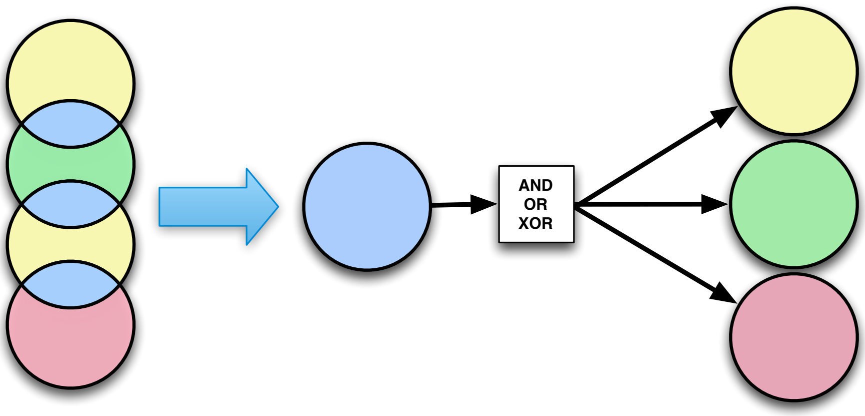


Figure 5.5: Commonalities in different patterns are extracted. Refinements are expressed in successors of the new, more abstract pattern.

Potential need for maturity change

After finishing the aggregation or decomposition process, the formulations within the refactored patterns should be reassessed by the community. Therefore, a change of the maturity state to *under consideration* is suggested. The already assigned evidence remains active but is taken into account again after ensuring the formulation quality of the restructured patterns.

5.3.2 Track Changes

Known logging of activities

A prerequisite for the traceability of the patterns' and library's development is a change log running unobtrusively in the background of which users are informed. This way, activity-related meta-information is stored that can gradually be shown together with the pattern or proposal processes that were described in preceding sections. The meta-information is always related to the current action taken, the user who performs the action, change description, the patterns as well as the relations that are affected. From the conceptual considerations presented in this chapter, the initial and extensible set of change actions is defined as follows:

DEFINITION 5.6: CHANGE ACTIONS
The set A of change actions consists of the following activities a_i:
- $a_1(p)$: Submitting pattern p to the library.
- $a_2(p)$: Accepting a pattern submission p.
- $a_3(p)$: Denying a pattern submission p.
- $a_4(p, p')$: Change of formulation of pattern p to p'.
- $a_5(p, l, m)$: Move of pattern p from hierarchy level l to hierarchy level m.
- $a_6(l)$: Proposing a new hierarchy level.
- $a_7(l, i)$: Insertion of a new hierarchy level l at position i.
- $a_8(c, p)$: Adding a comment c to pattern p.
- $a_9(r, p)$: Providing a rating r function to pattern p.
- $a_10(e, p)$: Proposing evidence e to pattern p.
- $a_{11}(e)$: Approving a proposal for evidence e.
- $a_{12}(e)$: Denying a proposal for evidence e.
- $a_{13}(r, p, q)$: Proposing a relation r between pattern p and q.
- $a_{14}(r)$: Approving a proposed relation.
- $a_{15}(r)$: Denying a proposed relation.

History of change actions

This initial set of chance actions can be extended in a concrete instantiation of the concept and altered depending on the project situation. An entry in the pattern library history change log is represented by a triple:

DEFINITION 5.7: PATTERN LIBRARY HISTORY
A Pattern Library History H is a multiset of tuples $c_i = (a_i(), u, d)$ where:
- $a()$ is the performed action from a set of change actions A.
- u is the user belonging to the set U registered to the pattern library.
- d is the change description given by the user u.

Reasons for development

The pattern library history concept provides reasons for decisions which is an important factor for the general understanding of the project-wide knowledge. Active specialists in particular project topics can be identified and asked for supporting the library with specific pattern formulations or by taking over an advisory role.

5.3.3 Suggestion of Pattern Sequences

Recommended patterns

The hierarchy concept of the pattern library possesses explanatory characteristic in the higher, more abstract levels. Patterns in this hierarchy level give information about rules, circumstances, restrictions and common practices that are usually set in a relatively fixed frame of the application domain. Processes can be adapted or optimized and rules can be changed. Yet, a larger potential for innovation and new findings is immanent in the middle and lower levels of the hierarchy since the patterns are becoming more concrete towards conceptions and realizations. The chance of alternative pattern combinations or mutually exclusive patterns grows in this region of the hierarchy. This way, the pattern library approach introduces a mechanism for *recommendations* for often used patterns, pattern combinations and reading paths through the library. The intention is to guide the reader, in addition to ratings of the individual pattern, through the existing set of patterns and to ensure a light-weight access to the contents. In case that the pattern collection grows, new users may be overwhelmed and demotivated by the amount of micro-documentations without assurance where to start and which patterns to explore to get an introduction to the topic.

Reading and application sequence

Therefore, two recommendation mechanisms are introduced: First, the *Reading Guide* intends to give orientation to new and inexperienced users that need to gain an overview of the project domain. The most important patterns need to be read first that provide the knowledge essences and mandatory rules that always must be kept in mind. Second, *Pattern Combinations* are formulated such that proven combinations of patterns are made prominent to users. Especially for decisions points, community feedback on made experience with different alternatives is considered as supportive. For both recommendation mechanisms, a definition for paths with weighted edges through the graph structure of the design pattern library is given that is based on the community's feedback and managed by the *Librarian*:

DEFINITION 5.8: PATTERN SEQUENCE
A pattern sequence s of all graph relations R within a pattern library is an acyclic directed sequence of weighted relations $w_1r_1, w_2r_2, ..., w_nr_n$ such that the initial pattern p_1 is a, and the terminal pattern of r_n is b, and for $i = 2, ..., n$, the pattern of p_i is the terminal pattern of p_{i-1} and for all $r_i \in R$ the relation with the highest weight w_i is chosen.

5.4 Management and Visualization

Feature coverage

Besides the conception of rules, roles, states and processes as described during the preceding sections, general demands and remarks for tool support regarding the management of the pattern library and visualization of user and structural activity were formulated and derived during the performed workshops. Related approaches that concentrate on different aspects of tool support are compared in Deng et al. [2005] who focus on extracting main requirements for tool support for researchers and user interface designers from the manifold of existing pattern language approaches. The authors differentiate between three kinds of tools. *Pattern catalogs* in general present the formulated patterns within their categorization. Mostly, web pages are used for dissemination. The second kind deals with *pattern management tools* that implement different ways for browsing, searching and visualization the pattern structure. Additionally, the latter two kinds of tools provide templates for pattern and creation and mechanism for structuring the pattern collection via categories as well as defining relations between the patterns. *Pattern-based design tools* give decision support to the current design problem at hand. The focus mostly lies on pattern from the human-computer-interaction (HCI) domain. This kind of tools partially allows the customization of patterns to the current application. The regarded tools were created between 2001 and 2004 and most of them are not publicly available anymore except for most web-based pattern catalogs. Still, the comparison enables the authors to create requirements for tool support for managing UI pattern collections. The authors address six problems that match to a subset of the requirements gathered in Chapter 4:

1. A common pattern form.
2. Customization of patterns.
3. Versioning of patterns.
4. Manipulation of forces.
5. Relating patterns.
6. Manipulation of pattern collections.

The evolving pattern library concept focuses on the aspect of *pattern creation, maturation and management* as described throughout this chapter. The rules of the process itself are conceptually defined but users take responsibilities for structuring the pattern library and assuring the quality of formulation and validation. In the following section, content management tasks are described together with needed features that support user acting in their assigned roles. Section 5.4.2 concentrates on needs for interacting with the pattern library such that the aspect of a *pattern catalog* is covered by the approach as well. Presentation and browsing of the library structure as well as the indication of changes, transitions and activities are important for the liveliness of the pattern library.

Features needed for creation and validation

5.4.1 Management and Maintenance

This section treats demands for management support of an evolving pattern library on a conceptual level. Especially for user roles that administrate and give advice to the contributions and structure, i.e., *Domain Experts* and *Librarians*, mechanism must be established to keep the overview of activities and states. For users in the role of *Members* and *Authors*, the state of their own contribution and activity of others must be shown. *Visitors* consume the knowledge represented by the library. Therefore, management tasks are not taken by this role. Visualization mechanisms (cf. Section 5.4.2) should provide easy access methods to the structure and contents in combination with an overview of recent changes.

Dashboard of activities

Members need to keep track of given comments and evidence they assigned during the collaborative shepherding process. Eventually, replies to them need to be taken into

Manage own comments

account in order to adapt or change submitted comments or evidence, respectively. During the workshops it was regarded as motivating to see whether comments were answered or evidence was accepted.

Manage own contributions

Authors must be able to keep track of their submissions and therefore need overviews of their prepared submissions and patterns that are submitted to the process. The current state of review must be clear to them and whether the submission has passed the first quality gate thus has been published to the collaborative shepherding process. Comments on submissions as well as change request should be easily accessible for them so that they can perform changes on their formulations. Mechanisms to react on comments, ratings and evidence by editing the submission need to be provided by the pattern library.

Overview on change and submission activities

Domain Experts need to keep the overview of newly submitted and recently changed submissions in order to accept or reject them. Tracking of a submission's development therefore becomes important. Evidence provided by the community must be organized and related to the corresponding patterns. Mechanisms to effectively manage evidence in a repository were regarded as useful to avoid duplicates. The overall decision progress of the *Submission Control Board* must be shown for each decision process. Based on a user's activity history, new candidates for domain experts should be identified and invited to the board as well. Overviews of suggested relations and demands for structural changes need to be available to the advisory board. Upon a structural decision, the *Librarian* is then assigned to perform the change.

Means for configuration

Librarians take over the core management tasks of the library system. They configure the quality gates at the establishment of the pattern library but also during its operation. All contributions, i.e., patterns, relations, comments and evidence, must be accessible such that the *Librarian* as system administrator has influence on all structural and content-related items of the library. Tools for monitoring and identifying neglected patterns support the sustainability of the pattern library.

5.4.2 Visualization of Structure and Activity

Library navigation

The concept of the evolving pattern library structures patterns via relations among patterns that in general express concretization of the knowledge. A graph structure according to Definition 2.2 is enforced in parallel with hierarchy levels within the graph that correspond to domain-related categories. The visualization of the pattern library as a graph covers the aspect of a *pattern catalog tool* and must provide an overview of the whole library structure, the relations between the patterns and the maturity state of an individual pattern. Browsing mechanisms along the graph structure as well as navigating within a detailed pattern view must be implemented. In addition, searching must be supported by dynamic filters or result lists for keyword-based queries. Pattern sequences should be made visible by drawing bolder edges in the pattern graph and therefore support the browsing of the whole structure along suggested paths.

Show activities

In order to address the liveliness of the pattern library, user and advisory board activities need to be shown prominently. This way, all users can perceive that the library is alive and are aware of others' contributions and actions. The visualization must mirror the own work of each individual user in a personalized profile that shows open tasks in a structured and easy to understand way. Newly submitted patterns, comments and evidence must be made prominent to pattern authors and members of the *Submission Control Board* who need to react and decide on new submissions. In order to support the tasks that need to be performed, overviews and task lists should support the orientation and task management. Changes regarding pattern formulations should be reflected by notifications or visual cues within the tool and graph structure.

Hints on mature and orphaned patterns

Concerning the maturity , a metaphor that expresses *increasing development* needs to be found. Structural changes and suggestions for new hierarchies or relations also need to be visualized in the graph structure and recorded in a change log. This way, the responsible members of the pattern library can provide feedback and decisions in short time. Orphaned patterns identified by the *Librarian* should be made visible within the graph structure and a prominent location such that the attraction of all members is drawn to these patterns again. Eventually, the increase of awareness of "work to be done" triggers the members' motivation to reassess the pattern and its advancement.

5.5 Conclusion

Pattern library as DAG with hierarchies

Based on the extracted problems with current pattern language formulation processes that are relevant for the application of the pattern concept as micro-documentations throughout the project scope, requirements addressing four aspects of the presented approach were formulated. The patterns are structured in a directed, acyclic graph structure according to Definition 2.2 given by Borchers [2001]. Hierarchy levels are introduced that categorize the patterns according to their degree of abstraction. Abstract patterns explain general rules, legal aspects and condition in the project domain and become more concrete toward the lower levels of the hierarchy dealing with processes, concepts and realizations. This way the library is able to make transparent the work performed by work packages concerned with domain analysis, application, interaction and process design as well as technical implementation and validation.

Collaborative pattern maturation process

The described*collaborative pattern formulation and validation* process starts with the submission of an initial pattern idea that can be published alternatively as an open problem seeking for a solution. Making use of community-based feedback, comments on the formulation are given and evidence is collected. The pattern maturity states each define a quality gate that needs to be passed in order to reach the next maturity state. According to the formulated process, approved patterns can be considered as well-formulated and valid. Yet, they remain within the process and are still part of the common discussion such that comments, reformulations and new evidence can be assigned furthermore. The process also monitors the age of a pattern and the community's activity on it. Eventually, a pattern needs to be revitalized by making the community aware of it again. In case that a pattern is no longer valid, it becomes outdated and unpublished from the library but remains in the library archive.

Pattern maturity

Different maturity states reflect the formulation quality and validity of a pattern. A generic rating function is used to define a formulation quality gate that covers the aspects of readability, understandability and appropriateness of a pattern. During the evidence gathering process, a pattern may turn into an anti-pattern and vice versa. An initial set of evidence factors is proposed that extends the generic evidence determination expressed by the *Pattern Maturity Indicator* (PMI). In a more advanced usage scenario, multiple indicator calculations depending on different weights can be defined for evidence factors. In addition, different PMIs can be defined per hierarchy level of the pattern library.

Role model

A light-weight role model assigns different rights to users of the pattern library. The role model defines *Visitors*, contributing *Members* as well as *Domain Experts* and *Librarians* that are responsible for ensuring the overall quality of the submissions and maintaining the library structure and its contents, respectively. Users in the role of *Librarians* additionally take care of configuring the parameters and rules of the pattern maturity process. *Domain Experts* are members of the *Submission Control Board* that takes decisions on new pattern submissions as well as the validation of provided evidence. *Librarians* form the *Structural Decision Board* that is responsible for change requests regarding the pattern library structure.

Formulation in natural language and pattern structure

Patterns are formulated in natural language following often used structures as proposed by Alexander [1977], Borchers [2001] and Schümmer and Lukosch [2007]. Additional fields like the explicit naming of evidence are added. The list of evidence differentiates between different types of evidence with different importance weights that either support or confute a pattern's validity. The concept of a pattern's origin is introduced as connection between the library and existing collections. This way, existing findings can be integrated into the project knowledge. For the applicability of the patterns in the project scope, adaptations to the project domain may be required. The anti-pattern indicator is optionally set by the pattern author when submitting a pattern and is automatically adjusted during the pattern maturation process depending on the assigned evidence.

Supervised structural development

The concept explicitly suggests a *dynamic library structure* that develops continuously. An initial proposal of hierarchy levels is given that can be aligned with the concrete project's domain. The structure equally underlies the dynamic development process meaning that new hierarchy levels can be introduced or existing ones can be merged if necessary. Patterns in all maturity states are associated with a hierarchy level and evolve over time. Relation to other patterns are added and changed over time based on the project developments. All changes are supervised by the *Structural Advisory Board* and performed by the *Librarian*. Tracking changes is an important mechanism to trace the development of patterns and the whole library structure.

Paths through the pattern library

Pattern sequences are defined to support the users of the pattern library. Arbitrary pattern reading paths can be defined for different purposes. New project members who need to gain an overview of the project domain should be supported by *Reading Guides* that mark a path of patterns that need to be read and understood first. Pattern combinations, especially in the lower levels of the hierarchy provide decision support when selecting from suggested solutions that are formulated as patterns and associated by XOR and OR semantics that extend the concept of pattern relations. AND semantics are introduced in case that information is distributed over different patterns.

Concepts for tool support

General design suggestions for tool support of the evolving pattern library approach were made with regard to managing patterns, relations and the structure with consideration for the tasks and activities for each of the defined roles. The requirements from the workshops regarding this aspect and related work in the field of tool support assist were taken into account. Overall, the approach covers the features of a pattern management tool and a pattern catalog.

Realization of the concept

The following chapter presents the iterative development of the technical implementation of the abstract EPL concept. Two refinements of the prototype were used to communicate the approach to the project members and to collect further suggestions for improvements on the concept as well as the developed pattern library platform. The aspect of visualizing the pattern library structure, maturity states and activities within the pattern maturation process are described in detail. A transfer of the EPL approach to the domain of service-oriented architectures validates the understandability of the concept itself and the applied visualization methods based on a usage scenario.

Chapter 6

The Iterative Realization of the EPL Concept

After the conception of the evolving pattern library approach was set based on requirements gathered in expert workshops according to the research agenda presented in Section 4.1, the key components of the approach were identified and an abstract concept was presented in the last chapter. The following sections describe the iterative implementation of a prototype used for refining is described. Practical usage workshops were organized to validate the prototype together with the concept.

Concept refinement and validation via prototype

Section 6.1 presents the first iteration of the EPL prototype that is based on an extensible content management system (CMS). With the help of the first prototype, the EPL approach could be introduced in the project BRIDGE that served as continuous testbed throughout the engineering efforts. In January 2012, a first pattern collection workshop within this project was organized followed by a usage period of the first iteration of the prototype spanning five months. At the end of the period, user feedback, suggestions and submitted patterns were evaluated and fed into the next iteration.

First prototype iteration

Based on the findings of the usage period and results from additional workshops that were held with the ARL Group at Reutlingen University in June 2012 and BRIDGE members in September 2012, requirements for refining the concept were gathered and fed into the implementation of the refined EPL prototype. Section 6.2 describes the final appearance, used technologies and developed components. A technical validation of the requirements concludes the section.

Refinement of the prototype

Section 6.3 describes the validation of the acceptance and applicability of refined version of the EPL prototype with business informatics students in the scope of a summer school event at Reutlingen University. In the scope of a two-days workshop, the prototype was used for teaching and documentation purposes in the domain of service-oriented architectures and cloud computing infrastructures. During the workshop sessions, the participants were introduced to the evolving pattern library approach and

Prototype validation

asked to interact with the system by adding and collaboratively reviewing patterns they were given as learning material. From these activities, lessons learned on the understandability of the concept and its technical presentation are drawn.

6.1 The First EPL Prototype

Introduction of the pattern concept and the prototype

The first prototype of the BRIDGE Pattern Library was used to introduce the pattern concept to the project and served as basis for the first pattern collection and concept refinement workshops. In order to introduce the notion of design patterns, an easy-to-read flyer was prepared to cover the basics of design patterns and their usage (cf. Appendix A). The flyer was disseminated to the project members to increase awareness about the library, along with an appeal to provide suggestions and feedback. Working efforts for developing the design pattern library were presented during two consortium meetings in order to encourage and trigger contributions to the pattern library. The early stage of the prototype encouraged participants to experiment with the implemented feature set and give suggestions for improvement. Similar to the findings reported by Dow et al. [2010], the participants were not afraid to suggest radical changes since the system appears as lightweight prototype that can be changes easily and that still misses sophisticate implementation mechanisms.

Seeding the library

As starting point for seeding seeding the library with patterns, rules and processes from the emergency response domain as well as first interaction and hardware prototypes were formulated as patterns. Along with the project development, all project members were encouraged to continuously submit any type of pattern or pattern idea to the library. As outcomes of the initial workshops on the EPL concept in September and November 2011, the following possible sources for patterns, open problems and pattern ideas were identified whereas contributors to the pattern library should always consider any copyright issues associated with content submitted.

- Experience and research conducted by the project partners.
- Arranged workshops with domain experts discussing design concepts and prototypes.
- Scientific publications: Papers, articles and theses written in the scope of the project.
- Online pattern collections. Careful incorporation and validation according to the project context may be necessary.
- Patterns derived from related fields.
- Patterns described in publications, reports and other literature sources like guidelines and principles.

6.1.1 Addressed Requirements

The concepts for the presented approach were described in Chapter 5 addressing the requirements for an evolving pattern library as formulated in Chapter 4. Table 6.1 shows the four identified components of the EPL concept with regard to the requirements. A "● " symbol indicates the full addressing of a requirement by the responsible component. "○ " means that the concept or implementation partially addresses the requirement but still needs refinement. The "⋆ " symbol indicates that other components provide additional support for a requirement although not explicitly demanded by the original architecture. In case that a conception or implementation is missing, the "⊖ " symbol is used. "−" means that the component does not tackle the requirement.

Functionalities and requirements

Table 6.1 shows that the derived concepts strongly address all requirements except for requirements R9: [*Authorship and Reputation*], R12: [*Structural Development*] and R16: [*Decision Support*]. However, first considerations were already formulated in Section 5.3. Additional considerations for further tackling these requirements are subject to future iterations of the approach.

Conceptual coverage

The technical realization of the first iteration of the EPL prototype mostly addresses the first set of basic requirements, i.e., R1: [*Availability of Knowledge*] to R8: [*Involvement of Community*]. Every pattern or pattern idea can be submitted to the pattern library by every member. A member of the *Submission Control Board* checks the quality of the submission regarding the completeness of the necessary fields according to the concept. After passing this step, the pattern is published in the pattern library as shown in Figure 6.1. For the first iteration, requirement R15: [*Visualization and Orientation*] was partially addressed by providing a grid-view based that shows all patterns.

Implementation covers basic requirements

The submission of a pattern is made via a form in which the necessary fields are indicated. This way, knowledge is made available very early to all visitors and members of the library. Refinements of the formulations can be made by the pattern author for his submitted patterns. Prepared views for recently added, changed and approved patterns are part of the visualization and arranged the main content area that either shows all published patterns, a single pattern's detail view, the submission form or explanatory articles that are linked to the menu items.

Pattern submission

During the submission process, authors are asked about the pattern's origin according to the conception. This way, a pattern can be indicated as originating from the project itself or an external library from which it can be included, eventually with necessary adaptations. A pattern's validity can be supported by providing evidence as free text meaning that references and links to any content are possible. Pattern submissions can be marked as anti-patterns. An indicator based on evidence as described in the concept is not realized.

Origin, evidence and anti-pattern indicator

Community involvement is principally addressed by allowing every registered user to formulate patterns. Feedback and comments on existing content can only be given via mailing the authors whose names are published together with the pattern. Regarding

Usage of standard roles

Overview of Features Addressing Requirements								
	Collaborative Pattern Formulation		Continuous Pattern Maturation		Dynamic Pattern Library Structure		Management and Visualization	
	Concept	Impl.	Concept	Impl.	Concept	Impl.	Concept	Impl.
R1: [*Availability of Knowledge*]	●	○	●	○	–	–	–	–
R2: [*Lightweight Contribution*]	●	○	●	○	–	–	⋆	⋆
R3: [*Early Participation of Stakeholders*]	●	○	●	○	–	–	–	–
R4: [*Liveliness and Extensibility*]	●	○	●	⊖	–	–	⋆	⊖
R5: [*Reuse of Knowledge*]	●	○	○	–	–	–	⋆	⋆
R6: [*Pattern Validity*]	⋆	⊖	●	⊖	–	–	⋆	⋆
R7: [*Anti-Patterns*]	●	○	●	⊖	–	–	⋆	⋆
R8: [*Involvement of Community*]	●	○	●	⊖	–	–	○	⊖
R9: [*Authorship and Reputation*]	○	○	○	⊖	–	–	–	–
R10: [*Role Model*]	●	○	–	–	●	○	–	–
R11: [*Transparency of Process*]	⋆	⊖	⋆	⊖	–	–	●	○
R12: [*Structural Development*]	⋆	⊖	–	–	●	⊖	⋆	⊖
R13: [*Structural Elements*]	–	–	–	–	○	⊖	–	–
R14: [*Pattern Maturity*]	⋆	⊖	●	○	–	–	–	–
R15: [*Visualization and Orientation*]	–	–	⋆	⋆	⋆	⊖	●	○
R16: [*Decision Support*]	–	–	–	–	–	–	○	⊖

Table 6.1: Overview of implemented features according to the defined requirements.

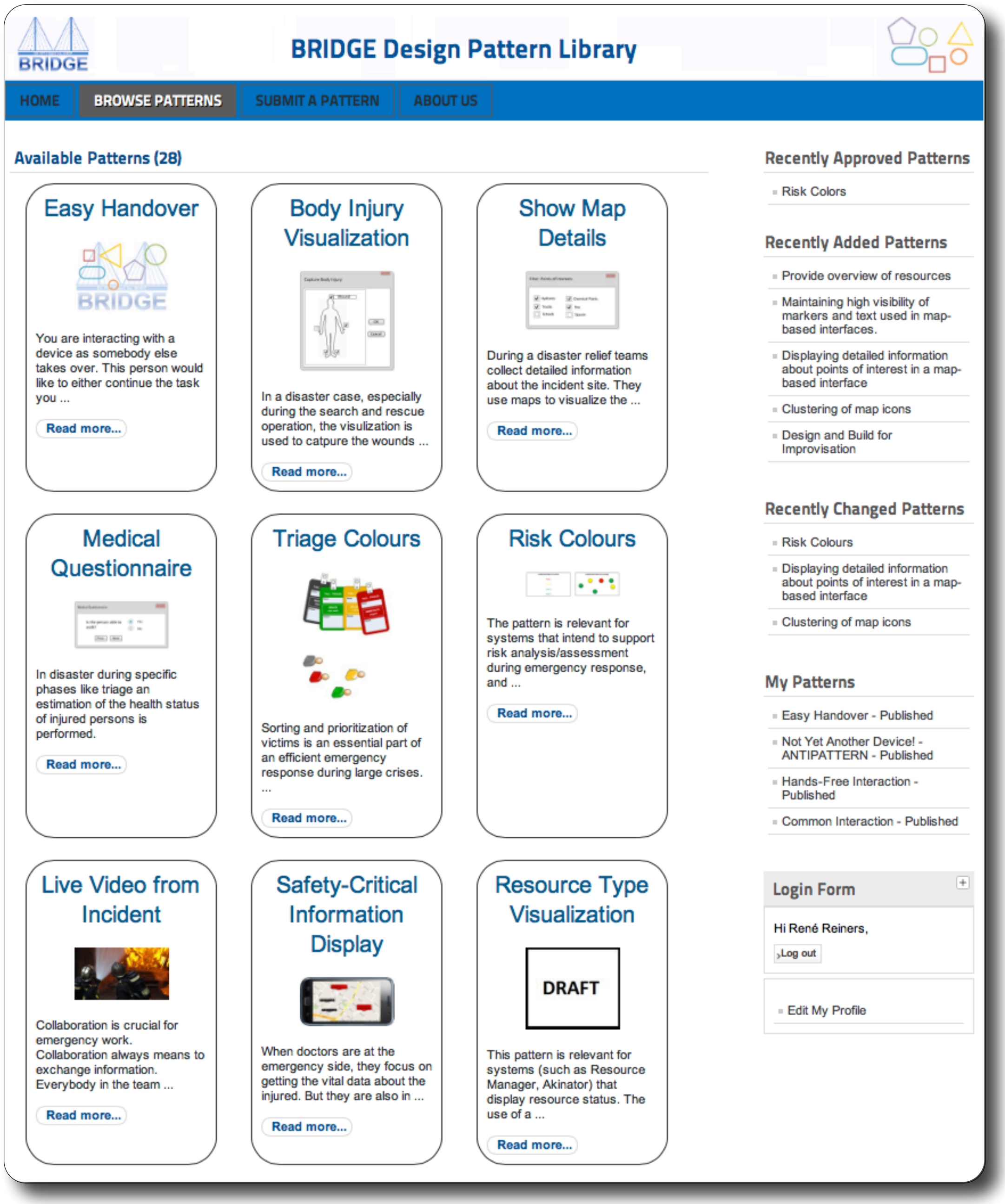


Figure 6.1: The view of all patterns as implemented in the first EPL prototype.

requirement R10: [*Role Model*], the envisioned role model was partially implemented and allows for the differentiation between visitors, registered users and administrators. For the implemented early pattern provision mechanisms, these roles were sufficient. However, for the more advanced mechanism for the increasing a pattern's maturity and validation, the role model needed to be extended.

6.1.2 Technical Infrastructure of the EPL Platform

Framework for collaboration

The technical realization of the EPL concept aims at providing a generic framework that enables all stakeholders in a project to collaborate and discuss on contributions as well as to manage the development of the library's structure. The project members can continuously fill the library with patterns discovered during the project work or existing from past experience.

Main features

The main functionalities of the final EPL Platform should provide means to:

- Browse the content of the library in different ways.
- Submit new patterns, pattern ideas or open problems.
- Provide feedback as comments and ratings on submitted patterns and the library's structure.
- Support or refute pattern proposals by assigning evidence.
- Propose suggestions for new associations between patterns.
- Determine the maturity states of a submitted design patterns.
- Visualize the liveliness of the platform, i.e., the maturation process of individual patterns, user activity and structural changes within the library.
- Provide administrative means for content management, structural changes and configuring quality gates and advisory boards.

Configuration and customization

The technical platform for the evolving pattern library is realized as a Joomla! component with a custom data model and mechanisms for pattern and association management. In addition, the rule and process model for pattern maturity determination and development is implemented within the component. Users in the role of *Librarians* act as system administrators and have access to the global configuration of the pattern component. For the approach presented in this thesis, basic settings are included as optional installation scripts that fill the database with the roles, maturity states and hierarchy levels as described in Chapter 5. The amount and denomination of the hierarchy levels is not fixed and can be adapted to the current project's circumstances. The proposed maturity states should be kept since the pattern maturation process is strongly connected to them.

The Joomla! CMS as Technical Platform

Separation of concerns

The contributions to the pattern library platform are interpreted as a composition of structured textual and multimedia content that is stored generically in a database and file system structure, respectively. For visualization purposes, the data is embedded into a customizable template that allows for styling and positioning of different views of the data. This way, a model-view-controller (MVC) strategy is pursued.

Reliable open source platform

For the technical realization, the open source content management system Joomla! was chosen. Profound technical competence with the system was gathered in previous projects concerning installation, customization and extensions development. The active open source community organized in the Joomla! Foundation[1] ensures continuous maintenance, development and security updates of the used framework. Because of the long-term development and community support of the Joomla! CMS, the technical sustainability of the platform is considered as reliable.

Standard core functionality

Joomla!'s main functionality is the handling of content as articles organized within hierarchical categories. The content is aligned to a defined template such that the author has less efforts concerning the layout and color scheme of the article but is mainly responsible for its structure and contents. The articles are prepared and managed in an administrative back-end where articles are categorized, edited and published after preparation. A role model assigns different right to edit, move and publish the content and administer the platform. The Joomla! Core framework already contains many basic features needed for the realization of the concept such that development can concentrate on the extension regarding the EPL concept. The relevant core functionality for the developed EPL platform is listed in the following:

- Easy management of content concerning publication and retraction mechanisms as well as meta data support.
- An extensible role model and access control lists for rights management. For example, authors prepare articles that editors review. Publishers are responsible for categorizing and making them available online.
- The technical framework relieves developers from standard programming tasks like login mechanisms, CRUD[2] database operations and form handling.
- Template support that allows the separation of concerns between content creation and visualization. Placeholders for different types of content can be defined and customized.
- User registration, login and session management that allow the restriction of specific contents and functionalities.

[1] http://joomla.org
[2] "CRUD" stands for the basic database operations Create, Read, Update and Delete.

- Support for third-party extensions that can be installed via a built-in mechanism and activated and deactivated on demand.

Open architecture

The architecture is extensible such that new components for pattern management, maturity assessment, user roles and groups, process rules as well as mechanisms for visualizations, browsing and searching can be realized and integrated into the framework. Joomla! supports various types of extensions, differing in complexity and purpose:

- *Components* represent the most complex kind of extensions and provide advanced management functions as well as custom data structures. They are administrated in the back-end of the Joomla! platform. For the front-end, specialized views can be defined that prepare the data in different ways. An example for a component is the article management as part of the Joomla! standard installation. It manages and organizes articles in the back-end and provides different views on single articles or summaries of article categories.
- *Modules* are used to prepare content from different data structures for special purposes in the front-end. For example, these can be menus, breadcrumbs, contact data or parts of articles. Module instances can be placed at defined positions in the front-end as an addition to the currently active component.
- *Plugins* prepare data before being displayed like inserting special HTML code or replacing variables with values before delivering the finally displayed content.

Available versions

The Joomla! framework has undergone several major development steps since its first release in 2005. At submission time of this thesis the Joomla! framework is published in version 3.1 which was released in March 2013. Since the work for the technical development has started long before the release of this version, the EPL platform is implemented to the last stable framework version 2.5.11 that is long-term supported until 2014[3]. The migration to the latest version can be performed with passable efforts as the compatibility between release strongly improved over the different releases.

Underlying technologies

The Joomla! CMS is implemented in PHP 5.3[4] whereas the content data as well as configurations and settings for the installed extensions are stored in a MySQL[5] database schema in version 5.0.4 or above. The Apache Application Server[6] since version 2.0 is needed as underlying runtime environment. Content is prepared and structured with HTML and CSS technology whereas the used version depends on the used template. JavaScript[7] is supported for dynamic content display. Besides the MooTools[8] and jQuery[9] libraries that are part of the standard installation, additional libraries can be integrated into the framework on demand.

[3]See download information at: http://www.joomla.org/download.html
[4]http://www.php.net
[5]http://www.mysql.com
[6]http://httpd.apache.org
[7]https://developer.mozilla.org/en-US/docs/Web/JavaScript
[8]http://mootools.net
[9]http://jquery.com

6.1.3 Lessons Learned From the First EPL Prototype

Initial pattern set

After introducing the EPL prototype during the first usage period starting with the pattern collection workshop in January 2012, the library was seeded with an initial set of patterns. Especially around the project review in April 2012, the activity increased again. Before the release of the refined prototype in June 2013, 28 patterns were formulated. From the published patterns, 22 were in the state *just created*, 2 patterns were *under consideration*, 3 of them were regarded as *pattern candidates* and 1 pattern was *approved*. It must be kept in mind that the maturity model was not yet fully derived and states were manually assigned by the administrator based upon discussion. The two states *just created* and *under consideration* were merged in a later iteration in order to facilitate the maturity process and reduce the number of states.

Abandonment of user roles, pattern device and pattern type

During the first conception of the EPL prototype in December 2011, the workshop group considered the introduction of *user role* and *pattern device* fields as useful to better convey the purpose of a single pattern to the reader. The hierarchy concept was still left out in this design iteration since it was regarded as too complex for the collection of an initial set of patterns. For this reason, pattern authors needed to specify a user role that predominantly applies the concepts described by the pattern. In the scope of the BRIDGE project, these could be, e.g., paramedics, firefighters, police officers or incident commanders. The field regarding pattern device types assumed that patterns would primarily be generated concerning system and device interaction. However, as the workshops turned out, patterns were formulated independently of devices and interaction concepts. In these cases, the fields *user role* and *pattern device* had no meaning and were left out. Similar issues were encountered with the field *pattern type* that was intended to differentiate between user interface, interaction design and application design patterns. Any other concept that was described by a pattern was assigned the value "other". In retrospective, these pattern classes were regarded as too narrow for the potential of the approach to use patterns as general knowledge containers that cover all kinds of project domain-related knowledge.

Describing concrete realizations

The set of derived patterns and discussion of possible future ones revealed that on the one hand, very general and abstract concepts are well-suited pattern candidates can be extracted from the documentation and reporting assembled by the domain analysis work package. On the other hand, very concrete realizations were available for interaction, implementation and validation work packages. These very concrete goals asked for introducing a kind of pattern that leaves the traditional technology-independent way of formulation in order to describe tangible realizations and applied technologies. Still, brand names and further specifics of the used components should be left out but minimal requirements for the used technology need be used in the pattern description.

Introduction of hierarchy levels

As a consequence to the encountered problems regarding different pattern fields, the concept of *hierarchy levels* was introduced to the EPL according to the concept described in Section 5.3. Originally left out during the conception workshops for the sake of quickly gathering patterns from current developments, the pattern collection workshop showed the inappropriateness of the surrogate fields *user role*, *pattern device* and *pattern type*. Authors were irritated during formulating concepts and discouraged whether their contribution really was an appropriate pattern since not all fields could be filled in a meaningful way. Thus, for the refined design iteration, the hierarchy concept was introduced together with a first proposal for hierarchy levels. User roles and pattern types are set implicitly by the hierarchy level and allow a more general formulation of a pattern covering more than only one user role and device type, respectively. In case an author is not sure about the hierarchy level, there is only one field for which he needs assistance. Alternatively, the community or advisory boards make the decision on the placement within the hierarchy.

Feedback on Visualization and Introduced Features

Survey group for visualization features

In the presented first EPL prototype, the visualization was kept very basic in order to start the process and to provide grounds for discussion. For the conception of the visualization for the refined EPL prototype, an anonymous survey among the members of the project BRIDGE was conducted in September 2012. 18 participants could be recruited from which 13 completed the online questionnaire that is shown in Appendix B. Participants working in the field of computer-science, HCI and physics are regarded as one group of 10 participants. 1 member from the emergency domain took part in the survey whereas the remaining 2 participants did not provide additional information on their background. 8 of the participants stated their experience in the field of design patterns as "expert", 2 participants as "good" and 1 as "average". 2 participants did not provide an answer. The composition of the participating group reflects the state of work at which mostly concepts from the domain analysis were transferred to system, application and architectural design. By trend, the majority of the project members have a background related to computer-science. Still, the composition of the survey group encouraged to further proclaim the application of patterns as communication medium and the pattern library concept as evolving knowledge repository.

Concerns on scalability

The general concept of the first EPL prototype and its basic pattern browsing capabilities were seen as understandable as shown in Figure 6.2. The qualitative feedback regarding this question expressed concerns on the scalability of the current version as assumed during the conception phase of the first iteration of the prototype. The need for more structure as well as advanced browsing and searching mechanisms was formulated as a strong demand and requirement for successfully using the system with a growing number of patterns. Therefore, the structure of the library is important to handle more patterns. Tags, categories and hierarchies should be integrated to organize different kinds of patterns and understand possible relations between them. Relations between different patterns should be implemented as links. Searching and finding appropriate patterns were regarded as necessary feature for the pattern library. The current representation did not allow the search for specific kinds of patterns. Structures and categories were missing.

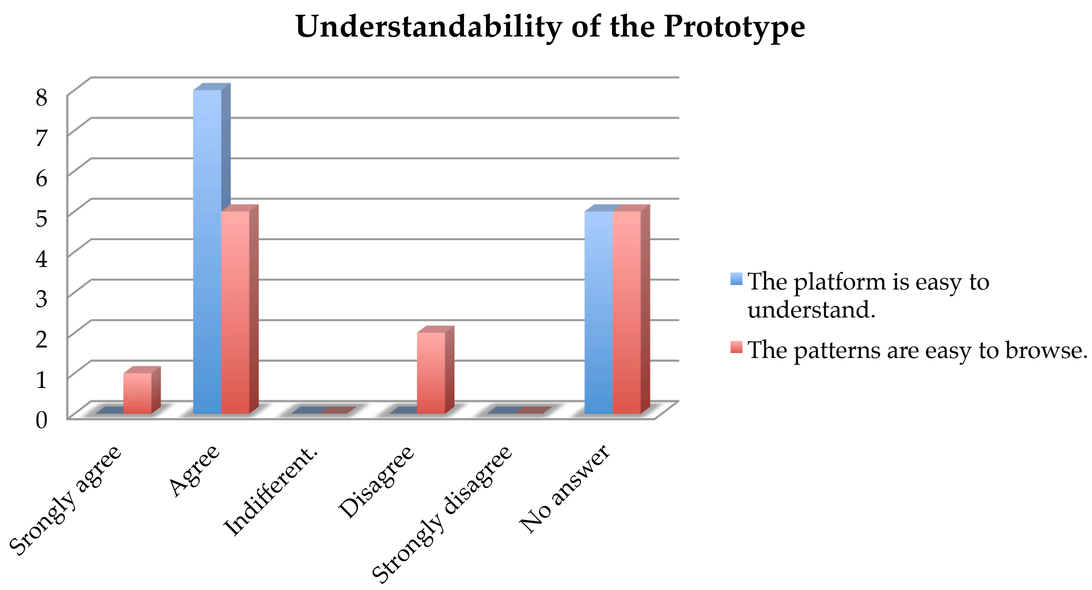


Figure 6.2: Easiness of the EPL prototype and browsing mechanism.

The patterns were sometimes seen as ”good advice”, best practices or requirements. This perceived mixture of concepts further encouraged the introduction of hierarchy levels for differentiating between general guidelines and concrete solutions to problems. The approach of using patterns as piecemeal growth of documentation must be conveyed by the representation of the structure. Still many participants expected the patterns to solve specific problems only.

Different views on the pattern concept

User activity and the liveliness of the whole pattern library were regarded as an important factor to show to all users in order to increase the motivation for browsing the content and contributing to the library. Different maturity states of a pattern should be visualized. Standard libraries usually consider all patterns as finalized advice. In the scope of the EPL prototype, pattern ideas should be easy to differentiate from approved patterns before reading the pattern details. Figure 6.3 shows the tendency towards the usefulness of showing user activities although more than half of the participants were undecided. The reason could be missing clearness of the concept behind the visualization of activities.

Liveliness and pattern maturity

Part of the envisioned concept of the EPL approach is the visualization of a pattern's maturity as well as the library's overall structure including the dependencies between single patterns. As shown in Figure 6.4, four different pattern metaphors were introduced. In the first conception, a pattern's maturation is shown via different states of the growth of a plant starting from seeds over seedlings, smaller plants up to a large tree. The second proposal made use of different moon phases to show the maturity of a pattern. An icon only showing clouds stands for an *open problem*, parts of a half moon

Discussion on metaphor

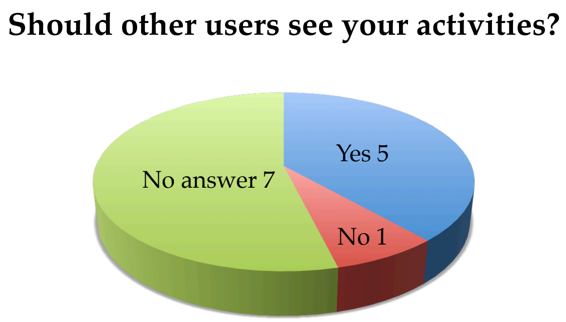


Figure 6.3: User activity should be shown but not shared via social media.

behind clouds and an emerging moon represent the maturity states *under consideration* and *pattern candidate*. A full moon completely visible is assigned to the maturity state of an *approved pattern*. The third concept shows different maturity states as different degrees of a city's development. Some users replied that villages are perceived as more agreeable and therefore more desirable. The last metaphor represented a mature pattern as a fully grown butterfly, the different maturity states were shown as a caterpillar, a pupa and a eclosing butterfly. Few participants stated that it may not be clear, why a moth should be worse than a butterfly.

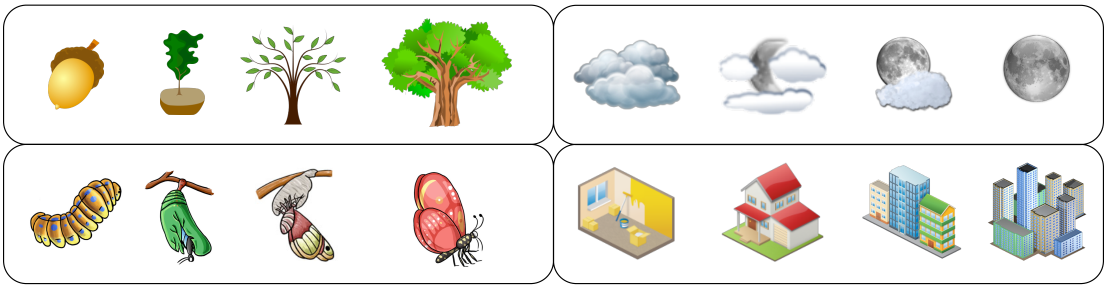


Figure 6.4: The proposed metaphors for visualizing a pattern's maturity[10].

Opinions on metaphors

The initial idea of introducing a metaphor for the structure and maturity states was seen as critical as the results presented in Figure 6.5 show. Depending on the user's personal attitude and experience, the applied metaphor could be misinterpreted. Due to the high degree of non-answers, the choice was made to try out a metaphor according to the ranking presented in Figure 6.6. From the survey, six participants provided a ranking for the different metaphors. Most votes were given for the plant metaphor and

[10] Plant life cycle icons taken from diverse contributors at http://openclipart.org.
Butterfly lice cycle images taken from http://www.butterflypictures.net.
Moon life cycle icons taken from the "Oxygen Icon Set" and
City icons taken from the sets "Large Home" and "Standard City"
see: http://www.iconarchive.com

the moon phases. Internally, the decision was made in favor of the moon phases since it was additionally motivated by the concept of *Harvey Balls*[11] that are predominantly used in the consulting business. They visually communicate qualitative information by circles at different filling levels. Concerning the technical infrastructure, the metaphor is exchangeable depending on the current projects preferences.

Figure 6.5: Opinions about the need for a visualization metaphor for a pattern's maturity.

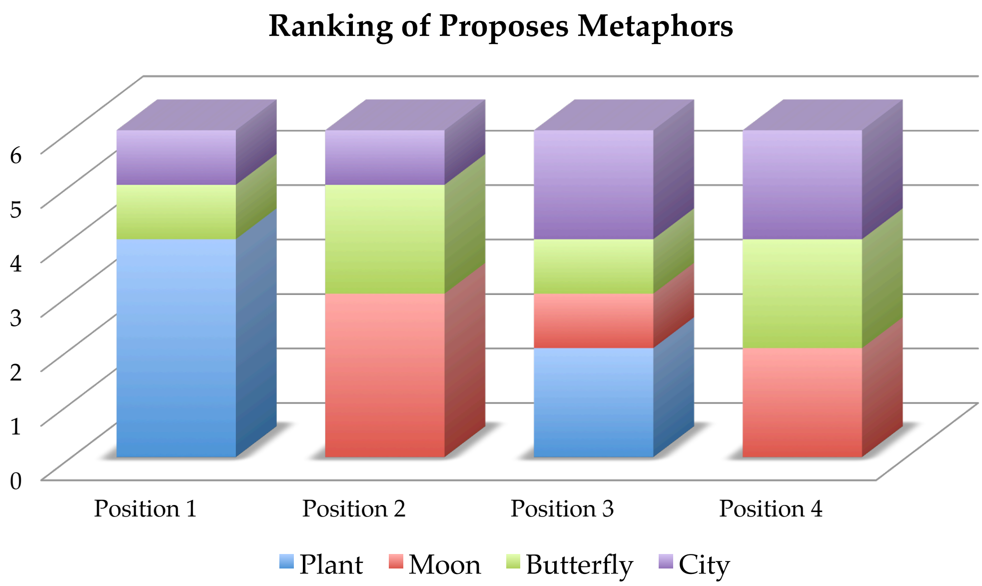


Figure 6.6: The results for ranking the presented metaphors.

[11]http://en.wikipedia.org/wiki/Harvey_Balls

Qualitative Feedback and Suggestions for Improvement

Feedback on pattern structure

Besides the collected patterns, encountered pattern formulation problems with regard to the proposed pattern structure and feedback on the visualization ideas of the first EPL prototype, qualitative feedback was gathered during the first pattern collection workshop in January 2012, at the end of the first usage period in June 2012 and via the survey on the current implementation of the first EPL prototype in September 2012. The qualitative feedback further influenced the conception of the refined EPL prototype, its visualization and the embedding into the general project workflow.

Project workflow

Concerning the process as a whole, the feedback from the participants suggested that the EPL approach should be embedded as an agreement into the general project workflow. If introduced to the project as optional addition to established tools, it may quickly be pushed aside by obligatory project tasks. However, the newly proposed way of documenting knowledge as patterns was considered as potential replacement of excessive reporting and assembly of project deliverables to a certain degree.

Interested work packages

Regarding the potential for participation, the initial discussions on the concept suggested that primarily work packages concerned with domain analysis, interaction and application design, validation and socio-ethical considerations may benefit the most of a pattern library. Work packages that concentrate on system architecture and exploitation tasks may partially be interested in reasons for decisions and knowledge that can be extracted from the library by following the patterns' developments. Still, the danger of loosing commitment to the pattern formulation approach was regarded as existent for all work packages within a project. The pattern formulation and validation process should continuously be "pushed" and driven further. Means to show the liveliness of the pattern library and activities of other users play an important role for keeping up the motivation for contributing to the pattern library.

Knowledge containers

In the beginning of the introduction of patterns as knowledge containers, the original meaning of patterns was partially irritating to members. It must be taken care of conveying the concept of patterns as generalized recommendation in a problem context. Often, a pattern was seen as a requirement that contains needs and constraints for engineering a solution. In another context, patterns together with their solution were interpreted as an "oracle" that provides a working solution based on a problem query. The answer was expected to be a single or a set of patterns that need to be applied in sequence to reach the solution. Both interpretations differ from the original meaning and usage of patterns as described in Section 2.1.

User needs as high-level patterns

One special connection between formulated requirements and pattern candidates was seen in the scope of the project BRIDGE: *User needs* describe general demands for a system conception and constraints that must be kept in mind when designing a system or application concept. These high level requirements formulate needs, constraints and parts of a solution by explaining what to avoid or to strengthen within a design. User needs may be transferred to more abstract high level patterns in the hierarchy.

Tasks for managing the contributed knowledge as well as the pattern library structure need to be separated from the actual contributions. These statements support the introduction of a role model in which users can perform different tasks ranging from reviewing contributions, placing them within the structure as well as changing the library structure based on current needs and suggestions. This way, authors and feedback contributors are separated from users taking over administrative tasks.

Support for role model

Some feedback was given on the generalizability of pattern formulations. In the original pattern concept, problems and solutions are formulated in a technology independent way. Regarding the application of the pattern concept as knowledge containers for a specific project's domain, it may become necessary to partially give up this quality of a pattern. It should be allowed to concretely describe acquired knowledge based on technical engineering, realizations and validations of achieved project results. The hierarchy levels that are introduced in the refined EPL prototype allow this kind of concrete formulation by introducing a special levels for this degree of concretization.

Concretizations in lower hierarchy levels

Additionally, the lifetime of knowledge was addressed during the discussions within the workshops. Questions arose whether spontaneous information with a short half-life should also be formulated as patterns. According to the original pattern concept, only long-lasting recommendations should find their way into the library. The EPL approach further allows the contribution of new information as comments and suggestions as well as evidence. Thus, short lasting pieces of knowledge should preferably not go into the pattern library but be considered as contribution to existing pattern formulations and validation efforts.

Patterns contain long-lasting knowledge

The concept of a pattern's maturity was considered as useful to be developed during the process. The mechanisms and rules behind its calculation need to be accessible and understandable by all participants. Otherwise, a pattern's maturation cannot be followed and eventually leads to a decrease in motivation. The visualization must convey the explanation of the concept setting without the need for additional documentation.

Pattern maturity concept must be clear

6.2 The Final EPL Prototype

The refined EPL prototype focuses on improving the collaborative formulation and validation of contributed patterns, pattern ideas and open problems. In addition, the semi-automation of the pattern maturity calculation and a structured visualization of the whole library are introduced. Most of the requirements that were formulated in Chapter 4 and addressed by the concepts described in Chapter 5 are implemented in the final EPL prototype that provides:

Focus on collaboration, validation and visualization

- Facilitated contribution mechanisms for new patterns, ideas or open problems as well as the provision of feedback on formulations and evidence for further validating the patterns. Discussion mechanisms allow the provision of comments for patterns in different maturity states.

- A realization of the proposed role model that differentiates between library *Visitors*, *Members*, *Authors*, *Domain Experts* and *Librarians*. The latter two roles organize themselves in advisory boards as proposed by the concept. The member size of the boards as well as rules for accepting group-based decisions can be configured depending on the current project environment.
- Concrete weights for aspects on pattern maturity calculation that are based on expert discussions during interviews for the implementation of the concept. The weight model is extensible and configurable on demand for each hierarchy level.
- An advanced visualization of the pattern library structure as hierarchical graph representation as well as a metaphor for illustrating the maturation of patterns and support for the awareness of other users' activities.
- Specialized views for each role reflecting the state of formulated patterns, given ratings and assigned evidence.
- Extended support for browsing and searching the patterns via the graphical representation.
- An extensible pattern hierarchy model that orders patterns by their level of abstraction.

Examples from validation in BRIDGE

The following sections describe the concrete configuration of the parameters for quality gates, advisory board sizes and hierarchy level names. The presented screenshots results from the state of the library in September 2013, after the final validation workshop that is presented in Chapter 7. This way, a concrete demonstration of the look and feel of the final EPL prototype can be given.

6.2.1 Structural Visualization, User Interaction and Role Model

Overall structure shown as DAG

According to the proposed hierarchy model and Definition 2.2 the visualization of the pattern library structure is based on a directed acyclic graph (DAG). Since the approach considers an dynamically evolving structure the graph does not have to be complete from the beginning. Missing relations are added throughout the process, others are never established. However, in a matured pattern library, pattern, there should be at least one path leading to each pattern. It represents the eligibility of the pattern and its context for the presented problem and solution. Associations between patterns are shown as directed arrows as illustrated by Figure 6.7.

Application of proposed hierarchy model

The EPL concept foresees the structuring of patterns via a hierarchy as used in existing pattern languages (cf. Section 2.2.2). This conception is supported by the qualitative feedback gathered with the first prototype in which missing structure was one of the main concerns. The refined EPL prototype adapts the hierarchy model presented in Section 5.3 that clusters the patterns depending on their individual level of abstraction.

Table 6.2 shows the proposed hierarchy levels reaching from high level legal aspects to more concrete concepts and realizations.

Laws and Ethical Considerations
General Processes and Concepts
Domain Practices
Application Concepts
User Interface Design
Interaction Design
Technology and System Design

Table 6.2: The proposed hierarchy levels for the pattern library.

Hierarchy description

The top level of the hierarchy includes legal aspects and ethical considerations, followed by general processes and concepts as well as domain practices. Application concepts, user interface and interaction designs are clustered in the middle hierarchies. These patterns hold explanations and reasons for existing and developed approaches. The lowest level describes issues concerning technology and system design. Still, patterns on this level should refrain from mentioning concrete technologies but describe solutions and facts on a generic base.

Structural visualization

The visualization concept follows the approach of hierarchically structuring the pattern library as directed acyclic graph is based on Scalable Vector Graphics (SVG)[12]. Associations between the patterns are highlighted when hovering over linked elements. An excerpt of the dynamically created pattern library graph of the BRIDGE pattern library is shown in Figure 6.7. The example shows the dependency between the concept of ”Risk Colors” and the ”Triage” process[13]. The latter is then extended by the ”eTriage Colors and Icons” pattern from which an user interface design is influenced described within the ”Body Injury Visualization” pattern. Users browse the pattern library by panning the provided graph. Tooltips show a summary next to the name. This way, users can preselect relevant patterns before reading the details.

Pattern details

When selecting an entry from the graph, the pattern’s details are shown on a new page as illustrated by Figure 6.8 for the ”Triage” pattern. The screenshot shows that the pattern’s current maturity state, hierarchy level and author metadata reside in the upper left corner. Below, widgets for rating the current pattern via Likert scales and evidence assignments are shown. Already assigned evidence that supports or refutes the pattern

[12] http://www.w3.org/TR/2011/REC-SVG11-20110816

[13] Triage: The process for sorting injured people into groups based on their need for or likely benefit from immediate medical treatment (see http://www.thefreedictionary.com/triage).

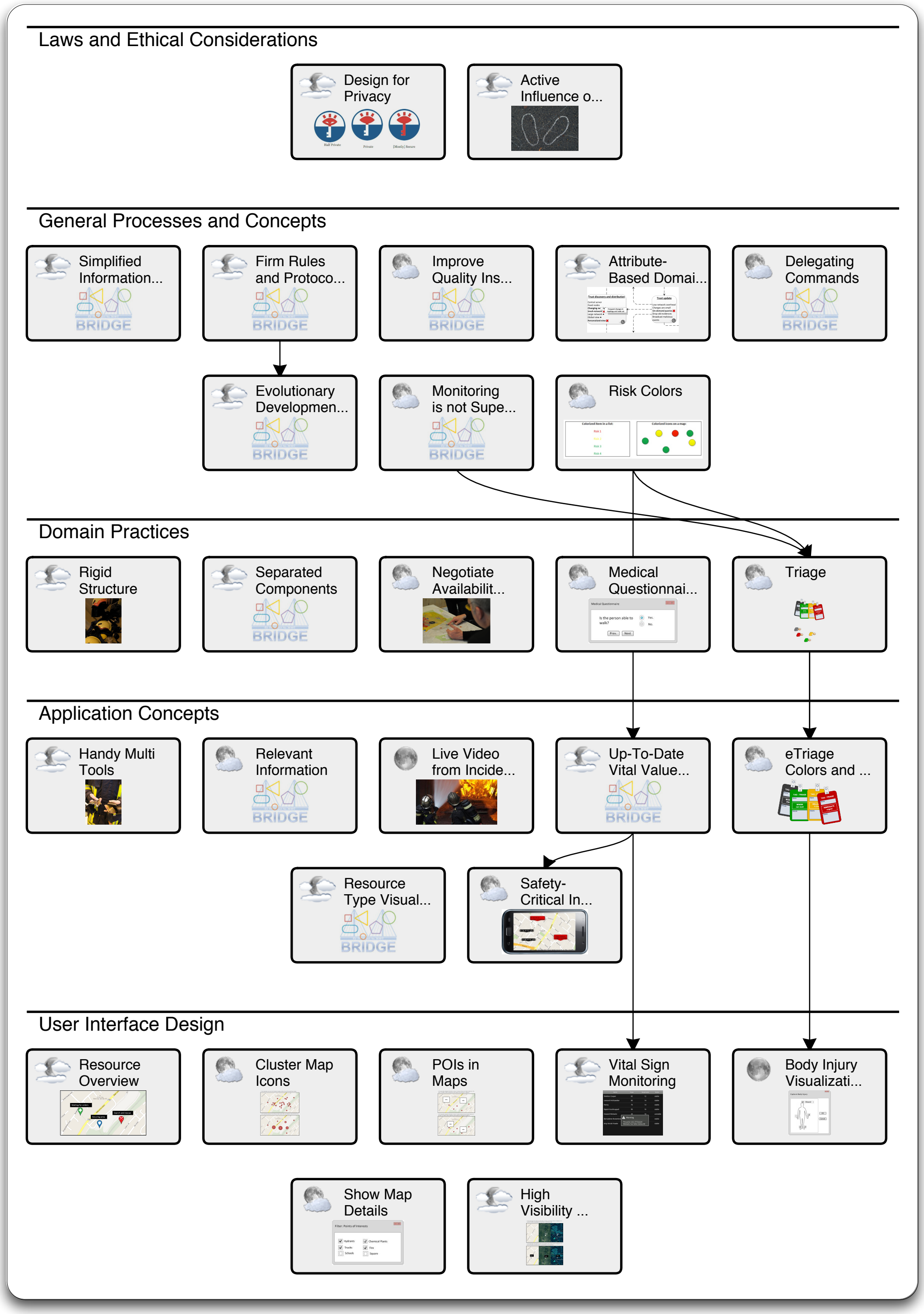


Figure 6.7: An excerpt of the BRIDGE pattern library graph visualization.

is listed. At the end of each pattern description, comments that allow registered user to provide feedback and suggestions and are collected per maturity state.

Scales and minimum votes

The rating function for a pattern's formulation quality is realized as three Likert scales that cover five discrete values. Each rating of a pattern is represented as a formulation quality tuple $q(r, u, a)$ where r, u and a hold the values for the degree of readability, understandability and appropriateness. Members can only provide one rating per pattern and change it over time. Only the latest voting is relevant for the formulation quality calculation. The EPL prototype's formulation quality gate is configured to require a minimum number of votes exceeding 20% of the registered members of the library. If each of the three aspects gains an average value above 3.0, the pattern reaches the next maturity state in which evidence is used for validation. In case that the number of members increases or later ratings go below the threshold, patterns which already passed the quality gate will not be reassessed automatically. Such a situation can occur in the scope of maintenance work that is initiated by the *Librarian* based on the pattern's age or the activity of the community regarding the pattern. This measure underlines the straight-forward approach of a pattern's maturation. Exceptional situations must be handled by *Librarians*.

Kinds of evidence and weights

Regarding the provision of evidence, users can either create new entries or select already existing evidence from a collection as shown in the screenshot of the modal dialog in Figure 6.9. For the assessment of a pattern's validity, the results from the workshops held in September 2012 are used as initial parameters. Thus, the evidence types and associated weights were extracted based on qualitative feedback from interviews as shown in part a) of Table 6.3. The results are presented as a proposal for a weight distribution of different kinds of evidence. Future iterations need to adjust the proposal according to the concrete project situation.

Different weights for lower levels

During the conception workshops of the refined EPL prototype, suggestions on a differentiation of the evidence calculation per hierarchy level arose. The EPL concept allows the formulation of different PMI calculations according to Definition 5.4. As a result, two PMIs were defined as shown in part b) Table 6.3. For the lower hierarchy levels, concrete examples, realizations and evaluations were considered as more important than publications that mostly regard abstract concepts. Examples can be found in related products, prototypes and validated developments from within the project.

Pattern submission

The refined EPL prototype provides specialized views for pattern contribution and iterative editing. Each member who wants to submit a new pattern formulation needs to fill in a pattern form similar to the one presented in the fist iteration of the EPL prototype. Obligatory fields are marked. In the refined version, the fields *user role*, *device type* and *pattern type* were replaced by the hierarchy concept. In case the author is not sure about the hierarchy level, he can leave this decision to the *Submission Control Board*. The maturity state of a newly submitted pattern is automatically set to *under consideration*. In case that no solution is formulated by the author, the pattern is declared as *open problem* to be solved in the future. The author can already provide evidence for the pattern

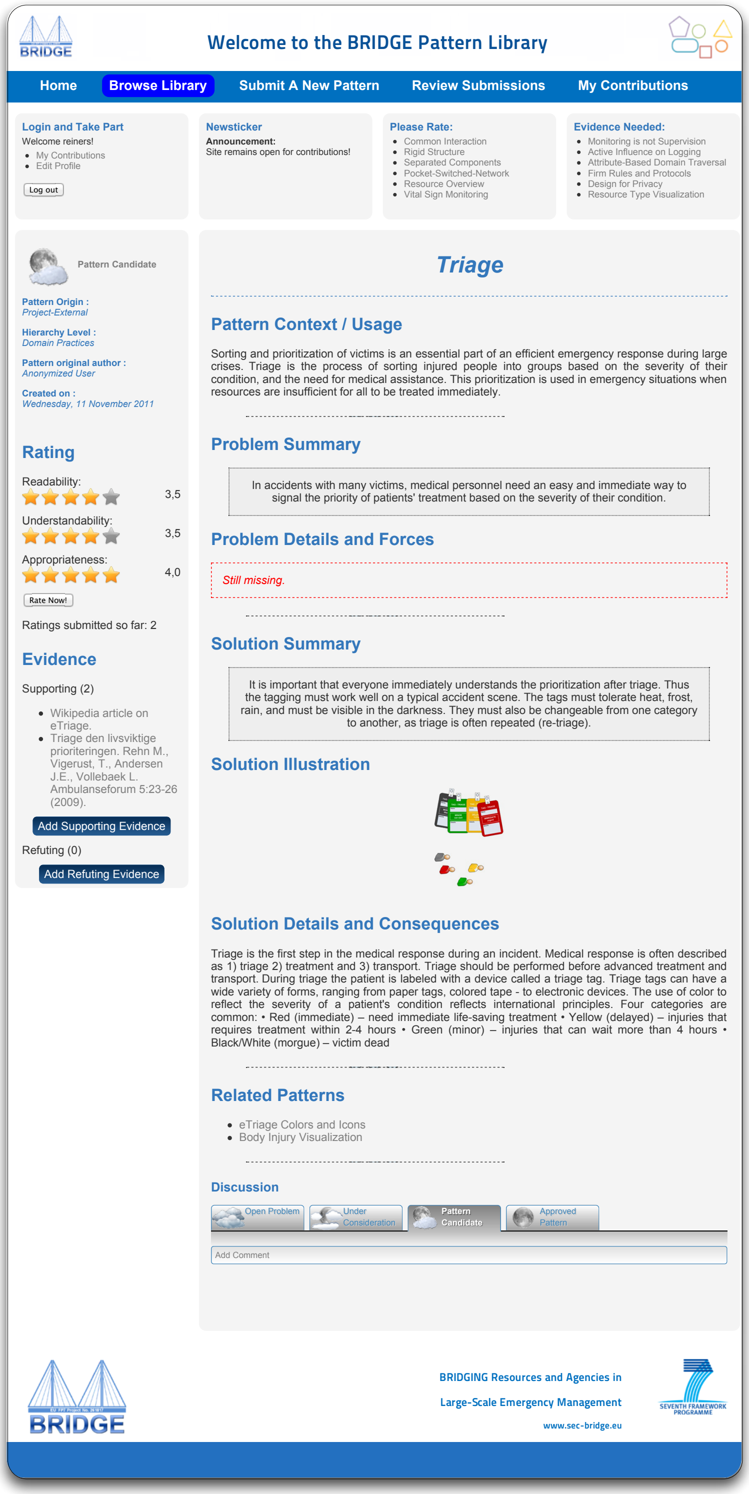


Figure 6.8: Example of the detail view showing the "Triage" pattern.

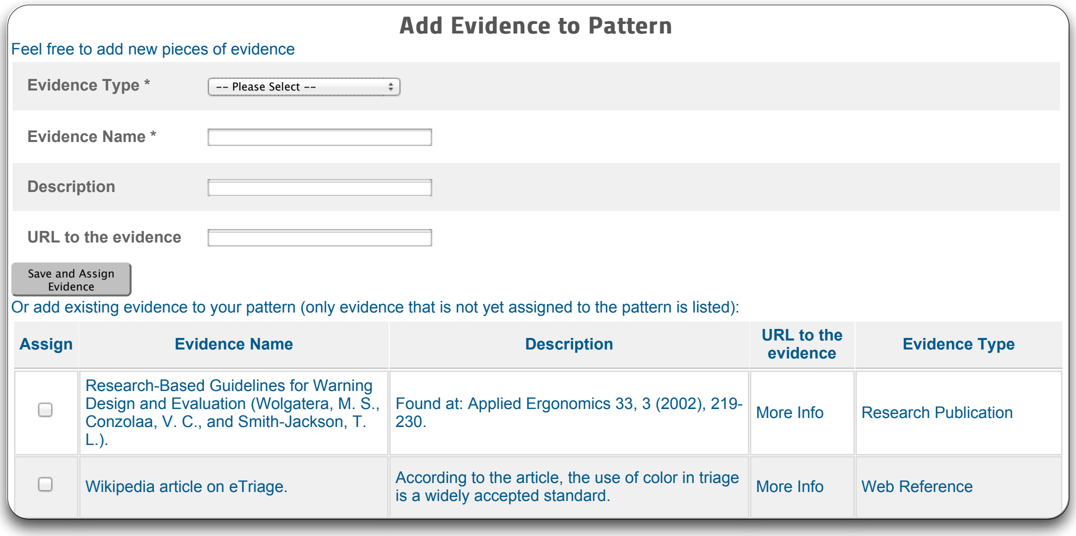


Figure 6.9: The form for assigning evidence to a pattern.

a) Standard Weights		b) Weights for Lower Levels	
Evidence Factor	**Weight**	**Evidence Factor**	**Weight**
Research Publication	.2	Product	.2
Ethnographic Study	.2	Existing Prototype	.2
Book	.15	Project Deliverable	.15
User Workshop	.15	User Workshop	.15
Project Deliverable	.1	Research Publication	.1
Product	.1	Ethnographic Study	.1
Existing Prototype	.05	Book	.05
Web Reference	.05	Web Reference	.05
Weighted Sum	**1.0**	**Weighted Sum**	**1.0**

Table 6.3: Proposals of weighted evidence factors for different hierarchy levels.

without influencing the maturity yet. First, the pattern and the assigned evidence need to be approved by the *Submission Control Board*.

Access to made contributions

Authors can modify their submissions at any time. A special menu entry that is shown after logging in provides an overview of the current status of made submissions as shown in Figure 6.10. It lists submitted patterns that are still in the reviewing process as well as accepted patterns. In the same way, assigned evidence is listed. Provided comments and ratings are summarized in special tabs.

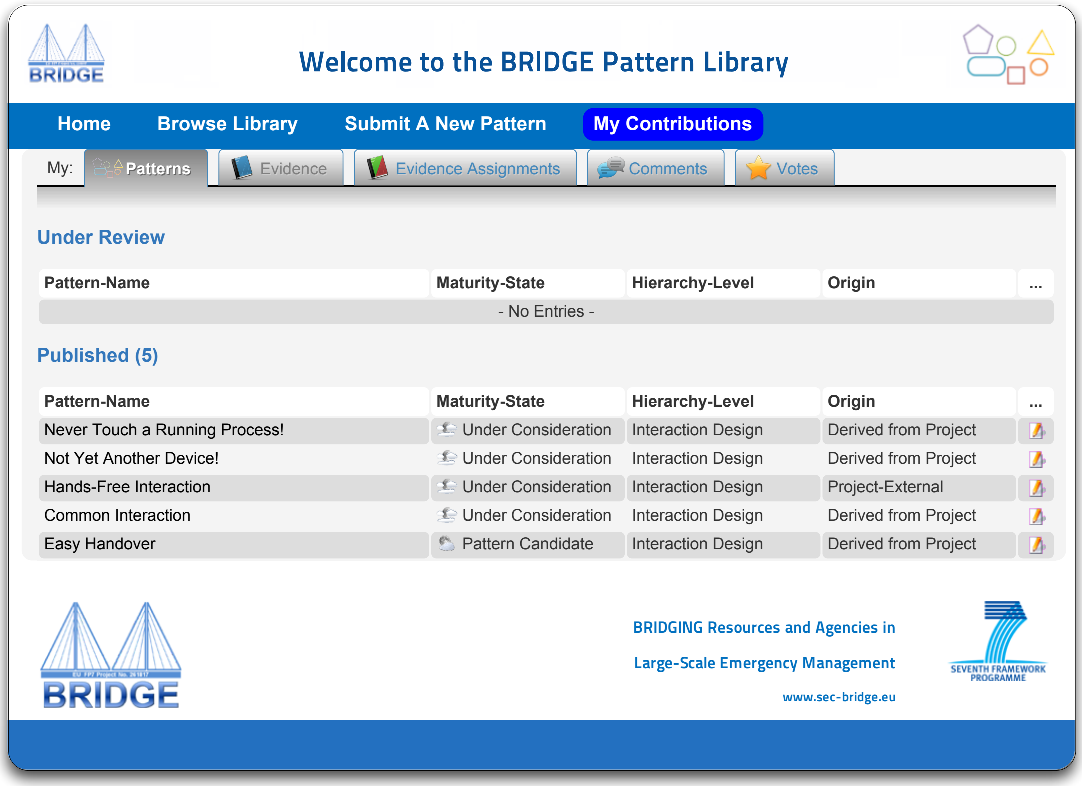


Figure 6.10: Overview of contributions made by *Authors*.

Submission review

Domain Experts see a special menu entry as shown in Figure 6.11. Users in this role can accept or reject submitted patterns and evidence as well as assignments. Additionally, "orphaned patterns" are listed in case the last modification or collaborative activity did not happen longer than a set period of time.

Extending the Joomla! Role Model

The Joomla! framework provides a standard role model including access control lists. It is extended by the newly created special roles according to the conception of the EPL process as described in Section 5.1.2. Similar to the Joomla! standard role of a *Publisher*, the *Librarian* is allowed to create, manage and put online articles created via the original CMS functionality in order to support the pattern library with additional content besides the patterns. Therewith, the standard roles of the underlying CMS remain

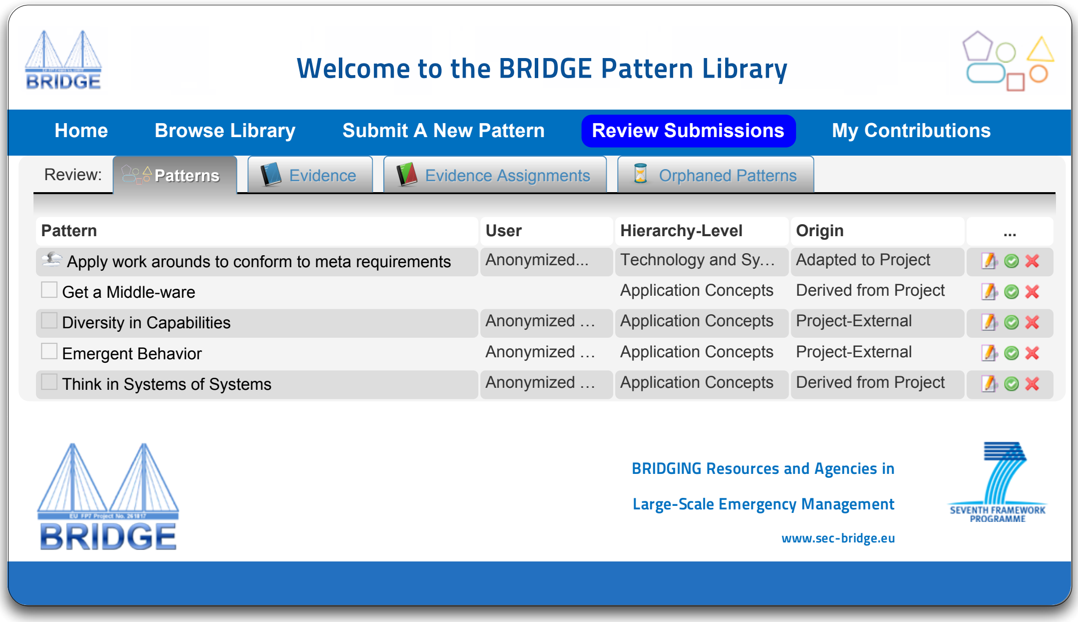


Figure 6.11: The view for *Domain Experts* for reviewing contributions.

untouched in their original functionality. This makes it possible to create additional content besides patterns that functions as publication and explanation material around the pattern library. All other Joomla! functions remain available and therefore are used for creating framing content such as welcome screens and explanatory articles. Only the new roles defined by the EPL component are allowed to contribute to the library by formulating patterns, giving comments and ratings as well as to assign evidence.

6.2.2 Developed Template and Widget Sets

Screen layout

Based on the common data model, the system architecture allows the flexible adaptation of the screen layout according to the current projects needs. In the following, the prepared views and widgets for the EPL realization are explained on an abstract level. A specialized template as shown in Figure 6.12 differentiates between several content areas. The main area of the template is reserved for the EPL component and displays, depending on the current task, the pattern library's visualization as a browsable graph structure, pattern details, forms for new pattern formulations or a modal screen for assigning evidence. Additional specialized views provide summaries of propositions, comments and decisions to be taken depending on the user's role. Around the main content area, several placeholders at the top and on the left side provide space for additional information and menus.

[14] Used stencils "Page Thumbnails" available at: https://www.graffletopia.com/stencils/409

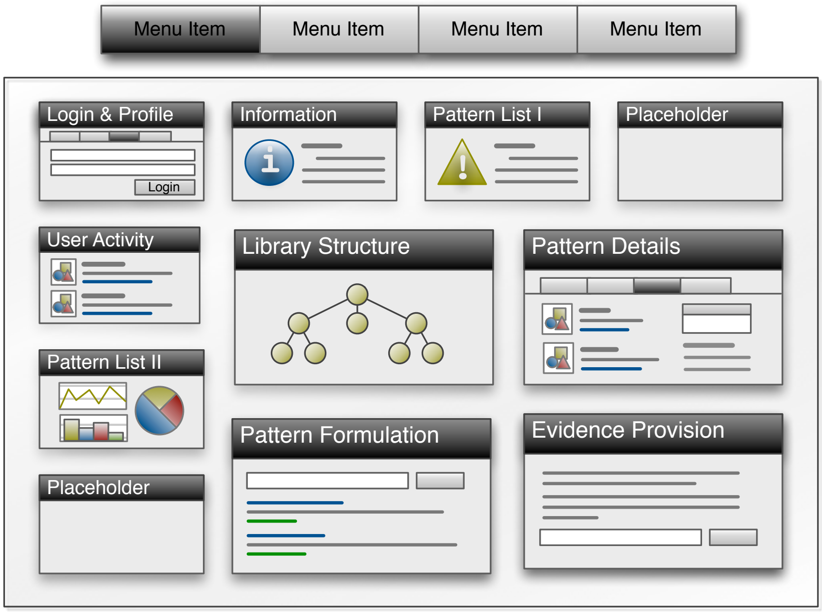


Figure 6.12: The conceptual template of the EPL platform[14].

Widget overview

Most widgets are realized as Joomla! modules that prepare the contents from the pattern- and user-related data structures. Figure 6.13 shows the implementation of the modules. Joomla! allows easily managing contents and placeholders such that future extensions can be added to the template or replace existing ones. The currently set of widgets that was developed for the refined EPL prototype covers:

- *User Profile Management*: Starting in the top left corner, users can either log into the system or create a new account that needs to be approved by a librarian. The integrated profile management allows for editing the stored personal data.
- *Newsticker*: Right of the login module, a newsticker shows latest announcements provided by domain experts. Users that are logged in see updates on their latest submissions. This way, they get to know about accepted pattern submissions and evidence assignments as well as given comments.
- *Need for Ratings and Evidence*: The next two items to the right list patterns that still need ratings or evidence. With every page reload, another arbitrary selection is presented.
- *Activity Stream*: On the left side of the template, user activity is presented as a list of events such as new user registrations, pattern submissions, provided comments, pattern maturity changes or evidence assignments. Additionally, the affected pattern, the time of change and the user who caused the event are shown. This way, more users should be motivated to contribute and perceive the liveliness of the platform.

- *Still Incomplete Patterns*: The widget shows patterns that need support regarding their formulation. Patterns whose fields are not filled completely yet are made prominent. This list should attracting the attention of authors and users who would like to help completing the formulations.
- *Hall of Fame*: This overview groups main activities performed by users such as submitting, commenting, rating as well as evidence creation and assignment. With the help of this list, very active users can be identified as future domain experts and eventually become a member of the *Submission Advisory Board*.
- *Who is Online?*: The last module shows the presence of other users that are currently logged into the system. This way, liveliness is shown in a more latent way.

6.2.3 Configuration and Management

Back-end administration

The Pattern Manager Joomla! extension provides different back-end views as shown in Figure 6.14. Tabs structure the component into views for submitted patterns, maturity states, hierarchy levels, pattern maturity indicators, pattern origins, evidence factors and the pool of assigned evidence. Librarians are allowed to administrate the whole pattern library structure, applied rules and its contents. Every item can quickly be published online or be taken off the platform.

Data management and rule engine

The component handles all CRUD operations to the database and updates related to user activities. Ratings, comments and provided evidence are handled by individual sub components. The same holds true for a pattern's history and structural development. Parameters that are relevant for advisory board sizes and thresholds for the quality gates are set by the *Librarian* and managed in the rule repository. On each pattern update, performed rating and evidence assignment, a system plugin that implements the rule engine for the pattern maturity process is triggered. It checks whether the pattern provides a solution such that it can be treated as a pattern *under consideration* or as *open problem*. In case a sufficient amount of ratings and the defined threshold are exceeded, the rule engine advances the pattern's maturity state. In order to transfer a pattern into the final state of an "approved pattern", the rule engine checks for the existence of a pattern maturity indicator defined for the current hierarchy level. In case that no special rule is defined, the global standard rule is applied.

Back-end vs. front-end

The back-end interface is primarily intended for administrative purposes as performed by the *Librarian*. For users in the *Domain Expert* role, the front-end provides interfaces for examining new contributions of patterns, evidence and evidence assignments as described later. The core data model related to the pattern management functionality of the custom Joomla! extension that is used by both views, back-end and front-end, is described in the following.

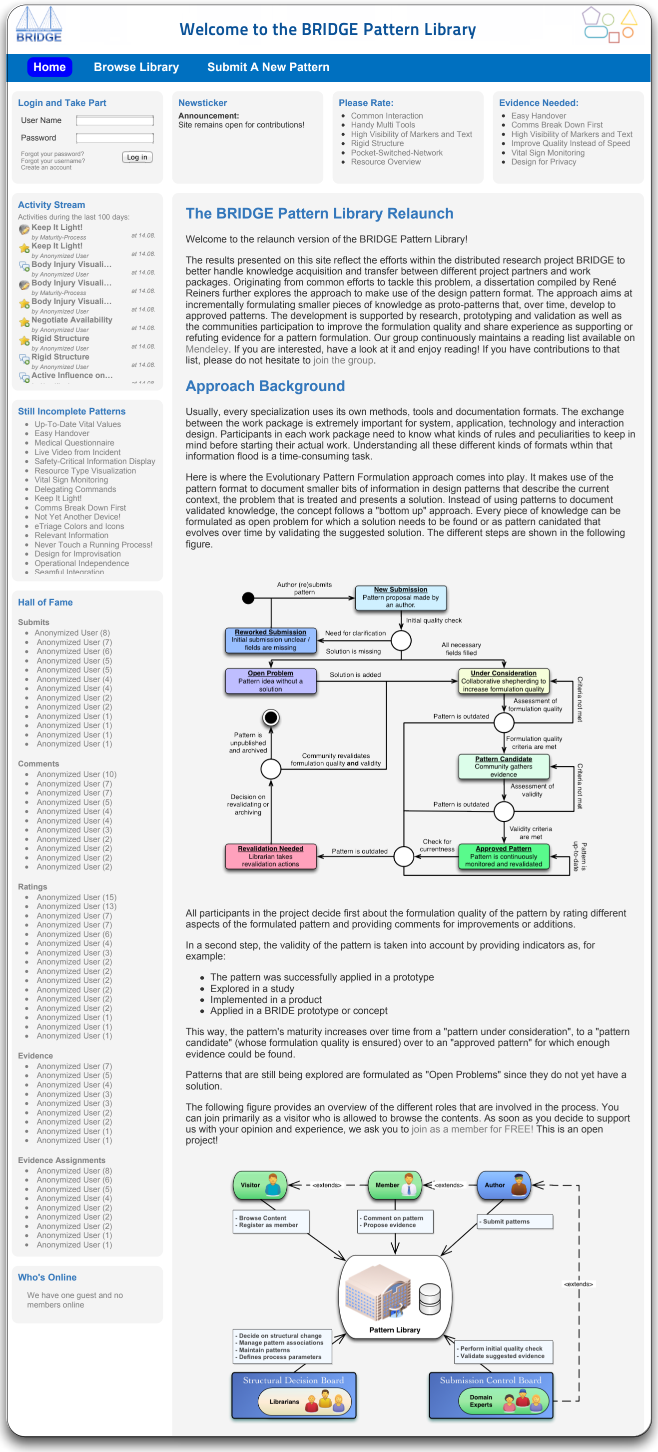


Figure 6.13: The start page of the BRIDGE Pattern Library.

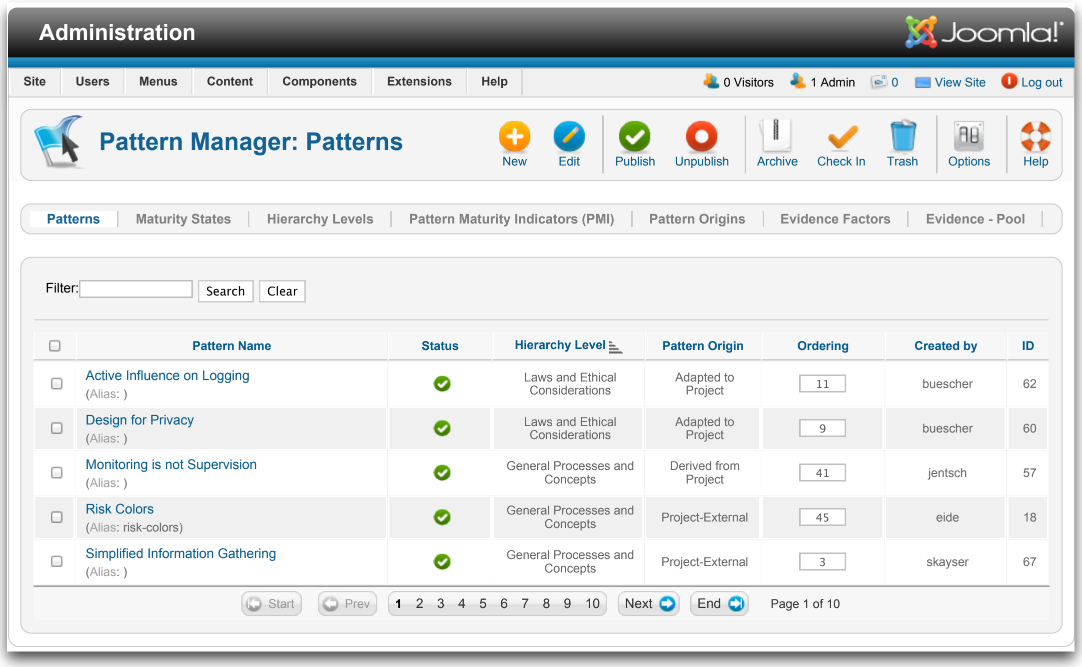


Figure 6.14: Back-end views divided by tabs for managing the structure and contents of the pattern library.

6.2.4 Data Structure

Reduced diagram

The developed pattern management component represents the backbone of the pattern administration and maintenance functionality since it holds the data structure for all involved entities. Figure 6.15 shows an entity-relationship diagram of the database schema of the EPL component. The diagram makes use of a reduced form of the different entities' attributes for clarity reasons. Entities and relations that are directly related to a pattern are drawn in bold style. It is noteworthy that the pattern entity has a reflexive relation to itself since patterns can be interrelated to others of the same or a lower hierarchy level within the graph structure according to Definition 2.2.

Common attributes

The primary keys that are defined by technical database-generated identifiers and attributes that are used for meta data information are not listed explicitly but incorporated into the data model. All entities have a common set of attributes that are used for tracking actions, authorship and feedback. The attributes of the common set are:

- *name*: A unique name of the entity.
- *ordering*: With this attribute, the relative position between entities of the same kind can be influenced.
- *state*: Publication state of the entry, i.e., "published", "unpublished", "archived"

or ”trashed”.

- *created_on*: Date of entry creation.
- *created_by*: User ID under which the entry was created.
- *modified_by*: User ID that performed the last change on the entry.
- *modified_on*: Date of the last modification.
- *checked_out*: Lock for avoiding parallel change on the same item.
- *checked_out_time*: Date of lock creation.

Specific attributes

Specific attributes of the different entities are left out in this overview since they are only technically motivated in terms of assigned icons, colors or aliases. The attributes regarding modification and creation dates as well as user ids associated with these events are a first step towards the establishment of a pattern history (cf. Section 5.3.2).

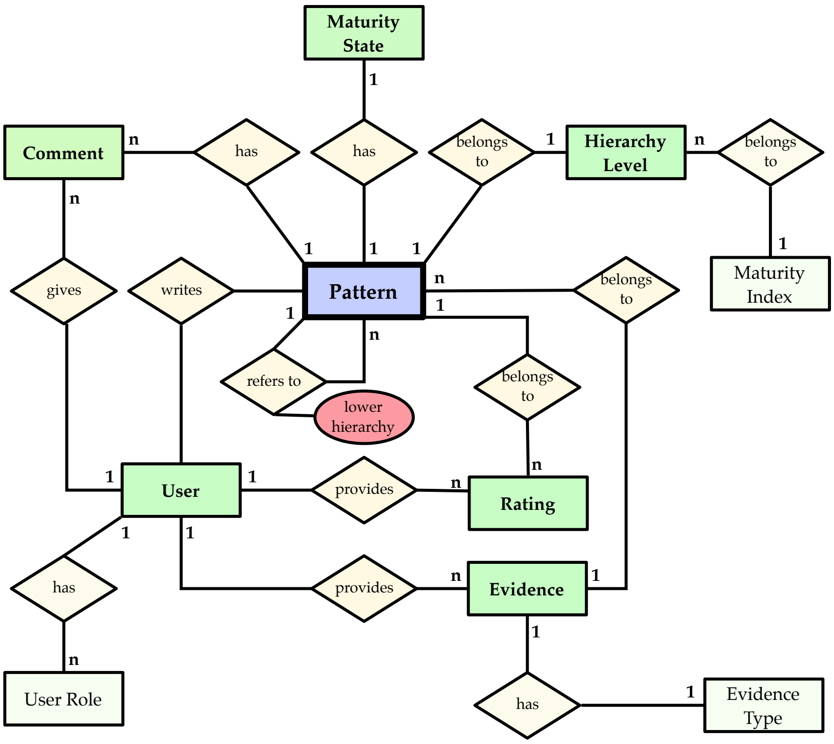


Figure 6.15: The reduced entity-relationship diagram of the EPL component.

6.2.5 Feature Validation

Feature of the refined prototype

The pattern maturation concept was modified for two practical application scenarios. For both instantiations, many of the elicitated requirements were implemented as shown by the feature matrix of Table 6.4. Similar to the feature matrix presented in Section 6.1, the issues that are fully addressed by the concept or implemented are shown as "● " symbols in the appropriate column in the matrix. A "○ " indicates that the requirement is mainly met by the concept or implementation. "⊖ " demands for tackling the issue in future iterations. A "−" symbol means that the component does not address the requirement since it is not mainly responsible for it. In addition, a "⋆ " means that a component partially supports the conception or the realization of the requirement.

The feature matrix shows that the majority of the gathered requirements is conceptually and technically addressed by the EPL conception and mature implementations of the four different system components of the pattern library prototype. In special cases, additional effort was put in the extension of originally non-responsible components in order to better tackle the requirement. Realizations primarily concentrated on the formulation and contribution processes in conjunction with mechanisms to manage and visualize user activities.

Technical walkthrough

According to the EPL conception described in Chapter 5, the progress of knowledge acquisition is reflected by the pattern concept as well as the availability of lightweight collaboration mechanisms. This way, all project participants are able to join and take part in the process whereas members with special roles are responsible for contribution management and structural maintenance of the pattern library. The current state of acquired knowledge is shown depending on the state of the according pattern's formulation. The provided and implemented maturity model includes rules for ensuring the formulation quality as well as the validation of a pattern. This supports the development of a pattern and the piece of project knowledge that it captures. All processes are made transparent to the participants. The notion of anti-patterns is introduced on a conceptual level. Evidence that refutes a pattern's statement is considered in the pattern maturity calculation and indicates the development direction of a pattern towards an anti-pattern and vice versa.

Extensible knowledge containers

Means for visualizing activity and managing contributions address the transparency of the process by the provision of a flexible template. Additionally, the graph-based visualization of the pattern library's structure as well as the implemented metaphor for a pattern's maturity state support this issues. All patterns can be extended by comments to the authors with suggestions for reformulations, evidence and ratings.

Transparency via visualization

Issues with regard to authorship and reputation models are partially addressed on a conceptual level. Further research and conception are needed in order to found a basis for realizations. Semantic annotations for structuring the pattern library can help to include semi-automated decision support. The prototype's technical architecture allows for future extensions and refinements.

Open issues

Overview of Concepts and Implemented Features Addressing Requirements								
	Collaborative Pattern Formulation		Continuous Pattern Maturation		Dynamic Pattern Library Structure		Management and Visualization	
	Concept	Impl.	Concept	Impl.	Concept	Impl.	Concept	Impl.
R1: [*Availability of Knowledge*]	●	●	●	●	–	–	–	–
R2: [*Lightweight Contribution*]	●	●	●	●	–	–	⋆	⋆
R3: [*Early Participation of Stakeholders*]	●	●	●	●	–	–	–	–
R4: [*Liveliness and Extensibility*]	●	●	●	●	–	–	⋆	⋆
R5: [*Reuse of Knowledge*]	●	●	○	○	–	–	⋆	⋆
R6: [*Pattern Validity*]	⋆	⋆	●	●	–	–	⋆	⋆
R7: [*Anti-Patterns*]	●	○	●	○	–	–	⋆	⋆
R8: [*Involvement of Community*]	●	●	●	●	–	–	●	●
R9: [*Authorship and Reputation*]	○	○	○	⊖	–	–	–	–
R10: [*Role Model*]	●	●	–	–	●	●	–	–
R11: [*Transparency of Process*]	⋆	⋆	⋆	⋆	–	–	●	●
R12: [*Structural Development*]	⋆	⋆	–	–	●	●	⋆	⋆
R13: [*Structural Elements*]	–	–	–	–	○	⊖	–	–
R14: [*Pattern Maturity*]	⋆	⋆	●	●	–	–	–	–
R15: [*Visualization and Orientation*]	–	–	⋆	⋆	⋆	⋆	●	●
R16: [*Decision Support*]	–	–	–	–	–	–	○	⊖

Table 6.4: Overview of concepts and implemented features of the refined prototypes according to the gathered requirements.

6.3 The ARL Summer School Pattern Workshop

Technical validation

Due to the cooperation with the ARL Group at Reutlingen University, the chance was taken to adapt the final prototype to a different domain and conduct an acceptance study of the EPL concept and its realization. Throughout a summer school event, the EPL approach was introduced and the final prototype was practically used during a two-days workshop. The ARL Group aims at applying patterns as communication medium for documenting and exploiting findings within partially distributed teams. Additionally, patterns are serve as teaching material regarding software architectures and architecture maturity assessment. The generalizability of the approach could be shown with regard to a different domain is seen. In addition, feedback on the EPL concept and its realization could be gathered from another point of view. Figure 6.16 gives some impressions of the introductory session and interactive workshop.

Figure 6.16: Impressions of the conducted ARL Summer School event.

6.3.1 Workshop Schedule

Approach introduction

The ARL Summer School was primarily focused on teaching and dissemination purposes for partners, researchers and students in the field of computer-science with influences on software enterprises and architectures. During the introductory lessons that were held on the first day for two hours, the pattern concept and the expressiveness of pattern languages were presented. As accompanying example, the Cloud Computing Patterns[15] were introduced. They are continuously updated and refined by Christoph Fehling and Dr. Ralph Retter who both have gained long-time experience in the field of software architectures and cloud computing. The website represents an excerpt of a larger collection that is formulated by Fehling et al. [2013].

Pattern format

The applied pattern format follows the usual structure including the pattern's name, a summarizing question, the context in which the pattern is used, a detailed problem description and proposed solution as well as an illustration. At the end of each pattern, consequences from its application are summarized and relations to other patterns are mentioned. Examples and references in literature support the pattern's validity. An iconic representation of the essence of the pattern helps to further convey its meaning as well as memorizing its content. Variations of its solution are briefly discussed if available. The whole pattern collection is organized in four semantic clusters describing cloud types, cloud service models, cloud offerings and cloud application architectures. Browsing happens via the listing or four menu items according to the clusters. The introduction closed with a summarized walkthrough of the pattern collection focusing on its different topics and application purposes.

Group composition

For the second part of the workshop, a practical session lasting for another two hours was set up with the aim of actively reading and reviewing the introduced patterns with the prepared pattern library prototype. The participants worked in teams of two. This way, they could reflect the contents of the introductory lecture and discuss the EPL platform they interacted with. In total, 34 master students of the business informatics discipline participated in the interactive part of the workshop. The age span reached from 22 to 28 years. Earlier semesters knew about the approach but were not as trained as their fellow students who participated in courses focusing the topic. In contrary, students from higher semesters were familiar with design patterns and actively applied them during their studies as references for programming and software architectures. Figure 6.17 shows the distribution of the knowledge level stated by each team.

Pattern formulation

All teams formulated one to two patterns with the help of outlines of the original Cloud Computing Patterns. Structuring headings were removed such that only plain text was shown together with the illustration of the solution. Thus, the patterns resembled sketches that first needed to be structured. Illustrations and icons were available in the online system. Participants were encouraged to incorporate new formulations or pattern ideas. The hierarchy level should be assigned at own measures or left out in case the participants were not sure where to place the pattern within the hierarchy.

Rating of foreign patterns

After the completing the submission of their own patterns, the participants should read other patterns in the library and rate them on the aspects "readability", "understandability" and "appropriateness", as originally prepared for the refined EPL prototype. This way, a peer-review process for the contributed patterns was established. Since the

[15]http://cloudcomputingpatterns.org

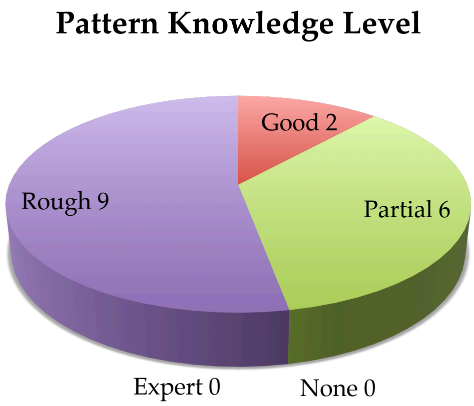


Figure 6.17: The distribution of the knowledge level on the pattern concept.

number of participants was manageable, the workshop chair was able to pay attention that different patterns were reviewed.

Evidence provision

For the third task, the participants assigned evidence to foreign patterns. Evidence factors consisted of references in literature, web publications and systems that are used in practice. The participants therewith tried out the collaborative validation mechanism and watched the change of pattern maturities based on the provided amount of evidence.

Survey and open discussion

At the end of the workshop, all participants took part in an anonymous survey that mainly focused on the understandability and traceability of the EPL concept as well as the visualization and interaction style of the prototype. The used questionnaire is shown in Appendix C. Throughout the whole summer school event, participants had the opportunity to give comments, suggestions and remarks as notes in a slip box. Finally, the participants were asked for qualitative feedback on the pattern maturation process and its realization via the library prototype in a 15-minutes wrap-up session that closed the workshop.

6.3.2 Pattern Library Setup

Template and advisory boards

By making use of the flexible template mechanism of the EPL platform, new color schemes and logos were assigned and needed widgets were arranged. The graph visualization and moon phases metaphor for different pattern maturity states was kept together with views of the community's activities. Patterns without ratings and those that are in need of more evidence were shown. The submission control board was occupied by the four workshop organizers as credible domain experts. They performed the initial formulation quality checks and assessed provided evidence. A *Structural Decision Board* was not set up explicitly.

Four-level hierarchy

The initial hierarchy model for structuring the pattern library was modified according to the structure of the Cloud Computing Pattern Collection. Thus, four levels were configured from. The highest level was intended to contain patterns dealing with different *types of clouds* as an overview of the topic, followed by *cloud service models*. *Cloud offerings* further explained the technical backgrounds and details behind the models. The lowest level of the hierarchy should handle the *cloud application level* for connecting components and services to the cloud technology.

Rating configuration

Ratings on the aspects covering a pattern's readability, understandability and appropriateness were given on a five step Likert scale. The minimal number of necessary votes was set to 3 and the average value for the quality criteria to 3.5. This configuration allowed for a reliable demonstration of the advancement of a pattern's maturity according to the process.

Contribution of evidence

The evidence model was adopted for easier usage during the short period of the prototype usage. The main aim was to let the participants interact with the prototype and explore the pattern maturation functionality quickly during the session. In order to make it possible that some of the patterns certainly reached the state of an *approved pattern*, every evidence factor was assigned the weight 1. A pattern maturity indicator ≥ 2 allowed the transition.

6.3.3 Lessons Learned From the Workshop

Time-lapse process walkthrough

During the practical part of the workshop, all patterns were entered into the library and the different steps concerning rating, commenting and providing evidence were performed. The participants were able to follow how different patterns gained maturity after submission to a *pattern candidate* and partially to *approved patterns*. Items that needed additional review were outlined by the system and thus, a subset of the entered patterns remained in the initial state *under consideration*. The setup allowed the participants to follow a time-lapsed walkthrough of the concept and portal functionality. It was explained to them that, in a real setting, contributions and modifications will potentially happen asynchronously and in larger time intervals. Project members will be spatially distributed. Therefore, the widgets that support the awareness of activities will probably become more important.

Understandability of the concepts

The understandability of the concepts was investigated with four questions as shown in Figure 6.18. The majority of answers states that the technical platform and the overall EPL concept that were easy to understand. 1 answer was indifferent regarding the EPL concept and 1 was not provided. Concerning the provision of evidence items, the offered evidence factors were understandable for 9 teams, 6 were indifferent about the types and 2 teams disagreed. The presented hierarchy levels that served more as pattern categories in the scope of the workshop were seen as understandable. Only 2 teams were indifferent and 2 did not provide an answer to this questions. The answers suggest to more clearly explain the meaning of pattern hierarchies and that their construction and naming must be performed very carefully. Answers on the understandability of the evidence classification suggest to prepare better explanations on the different factors and refine them iteratively over time.

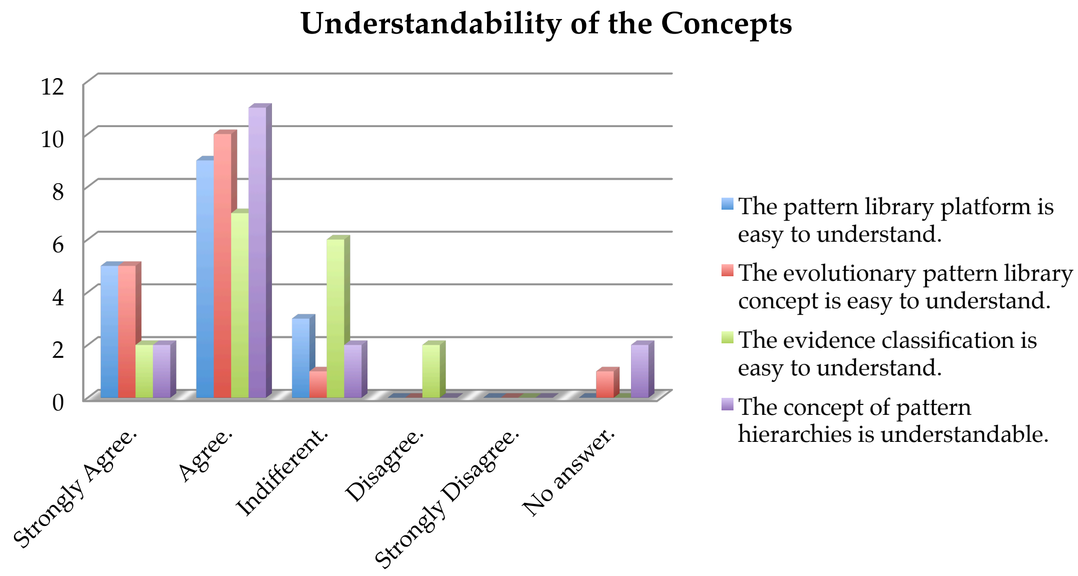


Figure 6.18: Combined chart of questions related to understandability of the EPL platform and approach as well as the evidence classification and hierarchy concept.

Roles and boards

The role model and advisory boards concept were considered as understandable by most of the participants as indicated by Figure 6.19. 1 team was indifferent and 1 disagreed to this statement. 2 did not give an answer. A similar situation with 2 teams being indifferent is seen for the concept of the advisory boards. Some more explanation on demand was suggested by qualitative feedback.

Clearness on the pattern maturity concept

Answers on the maturity concept indicate that, in general, the stepwise advancement of a pattern's maturity state was very understandable. Only 1 team was indifferent about the maturity concept and 1 team did not provide an answer. The results of the questions dealing with the maturity concept are shown in Figure 6.20. The overall idea of rules that are applied in order to improve a pattern's formulation quality and ensure support for its applicability were considered as understandable by 11 teams from which 1 strongly agreed to this statement. 3 stated to be indifferent and 3 others disagreed. As a consequence, the rules for the maturity advancement should be kept simple and provided as information on demand to ensure that every participant understands them or at least has the opportunity to look them up.

Maturity icons

Regarding the used metaphor, the moon phase icons that were leaned against the concept of Harvey Balls, was accepted and mostly considered as meaningful and easy to understand. Only 2 teams were indifferent and 1 team did not provide an answer. Maybe, the moon fitted well into the topic of cloud computing such that the connec-

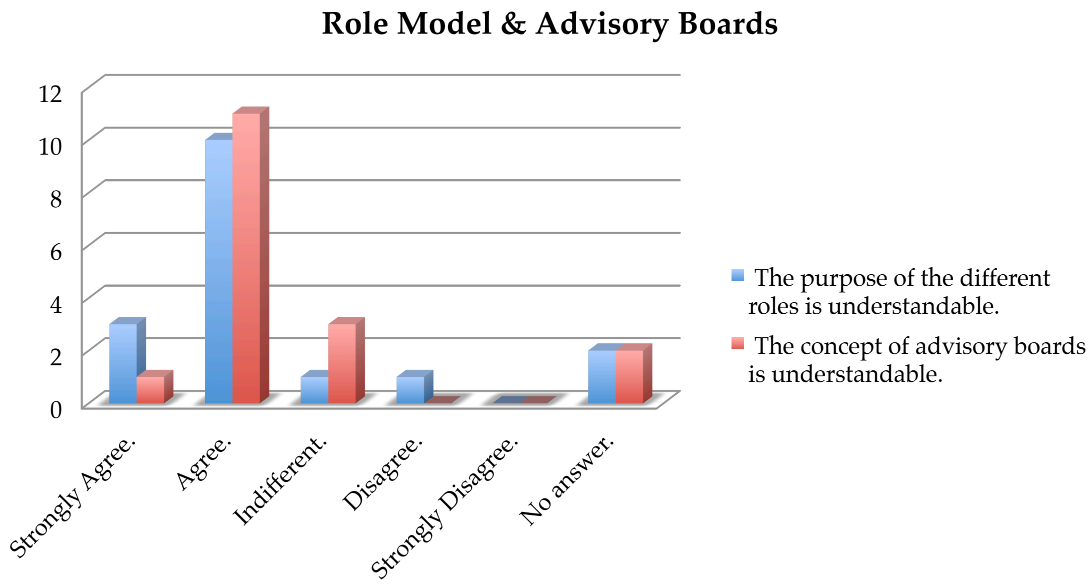


Figure 6.19: Understandability of the role and advisory board concept.

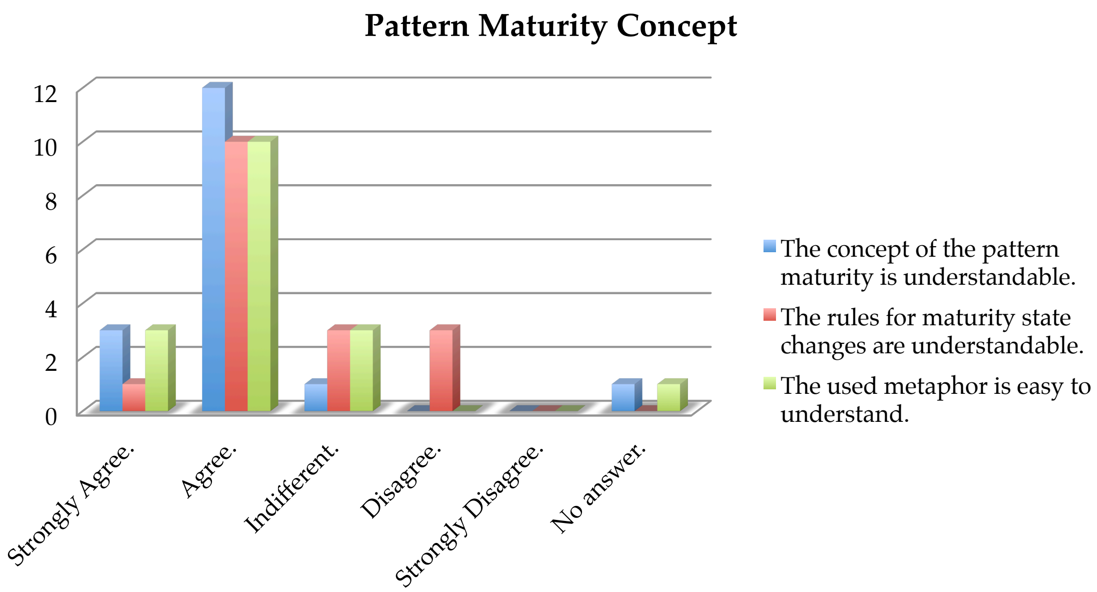


Figure 6.20: Understandability of the pattern maturity concept and rules for state advancements.

tion was made that way. Additionally, as stated during the interviews, the concept of Harvey Balls was known to most participants. Suggestions for improvements on the metaphor recommended the replacement of the moon with a "friendlier" sun symbol.

Usability of the platform

The answers to the next set of questions that are presented in Figure 6.21 deal with the general usability of the platform. Regarding the general usage and browsing the pattern library, only 3 and 2 teams were indifferent, respectively. 1 disagreed that the library was easy to browse. The answers support the general approach on providing information via few menu entries, informative widgets as well as an overview that presents pattern details on demand. Further improvements in cooperation with participants during project work should be taken into account for future usage scenarios.

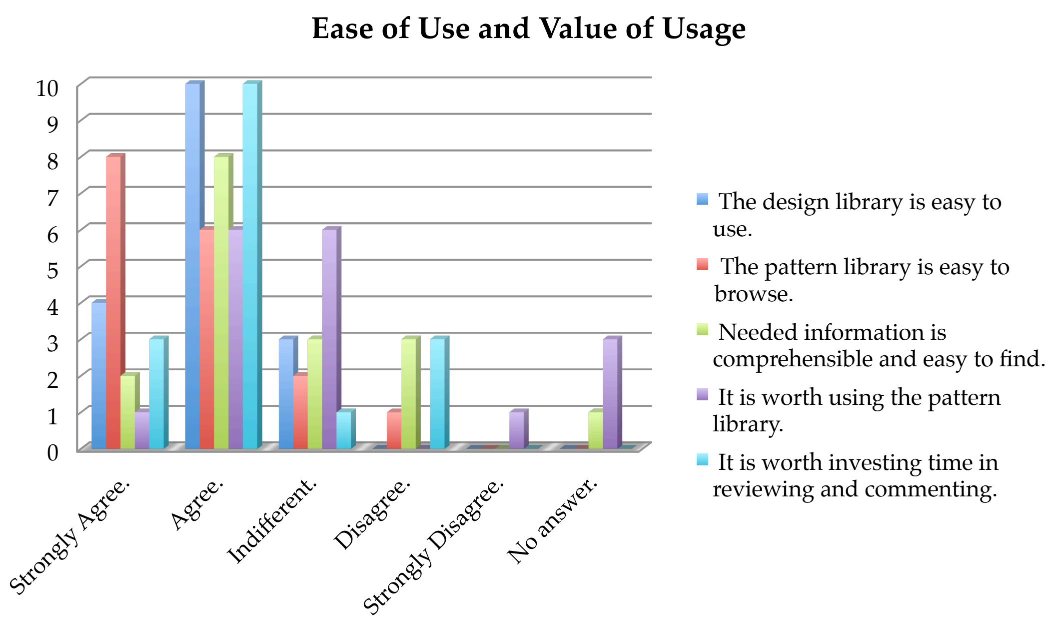


Figure 6.21: Combined results regarding the usage experience while interacting with the pattern library and estimations on its usefulness.

Interest in collaboration

Two questions asked about the participants' estimation whether they consider it worth using the platform and to invest time on reviewing and commenting the pattern. The answers indicate interest in the presented approach. Eventually, this estimation is also connected to the experience of the participants with the collection of knowledge as patterns during semester courses and the presentation of the perspective of the EPL approach in the introductory lesson.

Ease of making contributions

After making themselves familiar with the general usage of the pattern library, the participants contributed patterns, ratings and comments. Figure 6.22 shows the results of questions dealing with the different kinds of contributions. Entering new patterns to the library was not considered as difficult. Only 1 team was indifferent about this aspect. The provided rating mechanism on patterns was even more accepted. 12 teams strongly agreed to its easiness of use. Again, only 1 stated indifference. The submission of evidence and comments was considered as less easy, as indicated by 1 indifferent and 2 answers each disagreeing and strongly disagreeing on this aspect. Statements and reported misunderstandings during the workshop suggested to improve especially the evidence provision and internal review mechanisms.

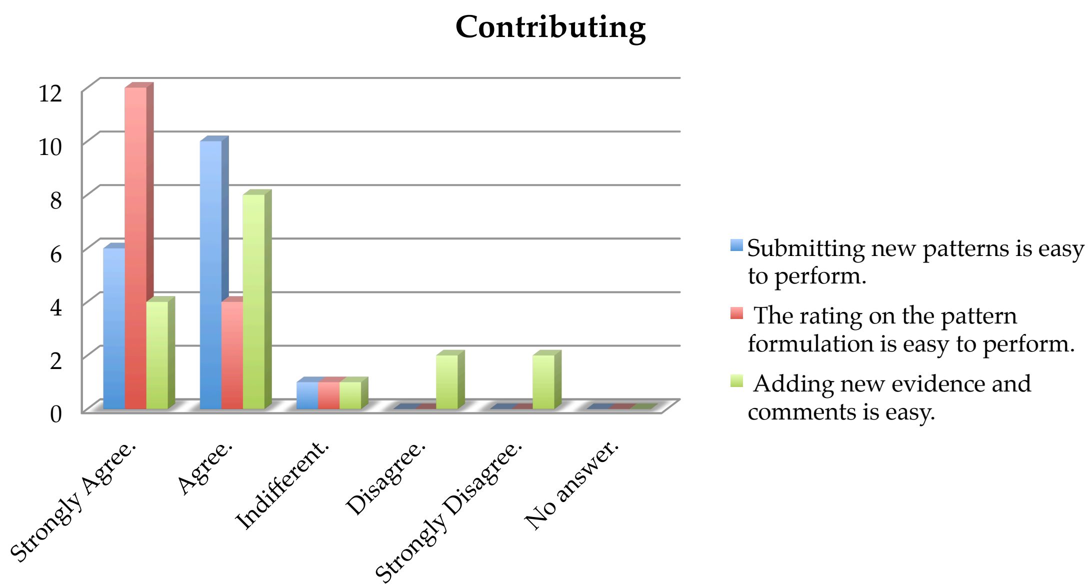


Figure 6.22: Answers on the easiness of contributing patterns, ratings and comments.

Overall visualization

In the scope of the ARL Summer School Study, the overall presentation of the pattern library, the pattern graph structure as well as the pattern details were considered as good by 11, 12 and 13 teams as shown in Figure 6.23. Only 1 team was indifferent about the general presentation and 3 disliked it. Similarly, 4 were indifferent regarding the visual appeal of the pattern graph structure. Pattern details were absolutely disliked by 1 team and disliked by another. 1 stated to be indifferent about the detail view. The remaining 14 answers stated satisfaction with the presented design.

Liveliness and motivation

One concern of the technical realization of the EPL platform is to keep users and authors informed about activities and contributions. In large-scale projects, members may be spatially distributed and work with the library asynchronously. Additionally, motivation to work with the library needs to be held up by avoiding the impression that the individual user acts in an isolated way and his contributions are not perceived by others. For this purpose, different kinds of widgets were designed and added to the library template. The questions regarding the liveliness of the pattern library asked whether the platform appropriately shows activities. Figure 6.24 illustrates that 10 teams agreed on this aspect. 5 were indifferent and 2 did not provide an answer. The widgets themselves were considered as understandable by 2 strongly agreeing teams and 11 agreeing ones. 2 answers stating indifference, 1 disagreement and 1 not given answer indicate

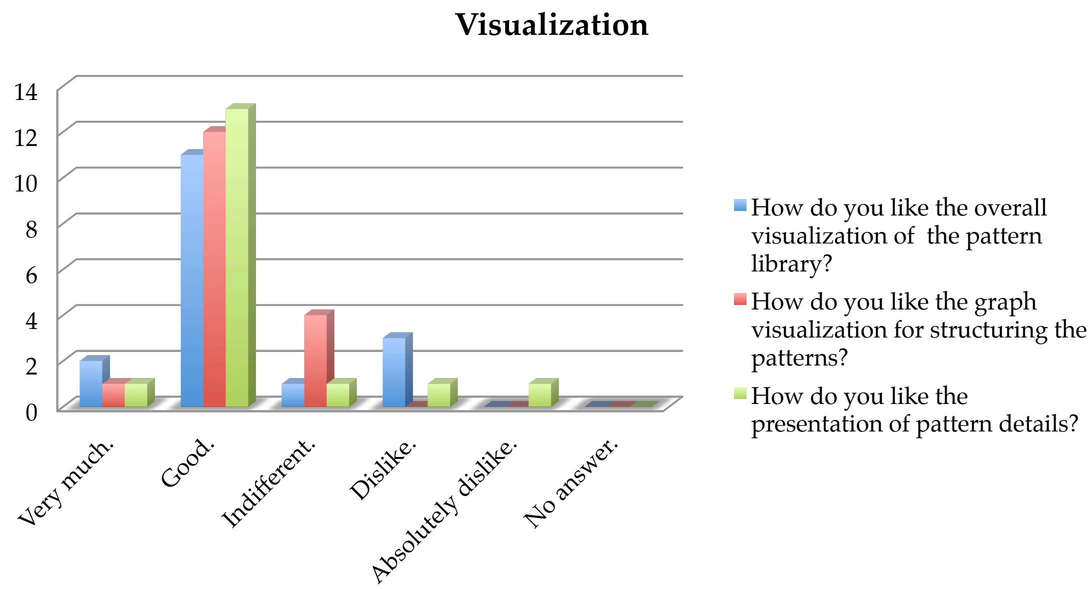


Figure 6.23: Feedback on the overall visualization, the pattern library graph structure and the pattern details.

that there is room for iterative improvement during project work but there are no urgent design flaws which need to be reworked.

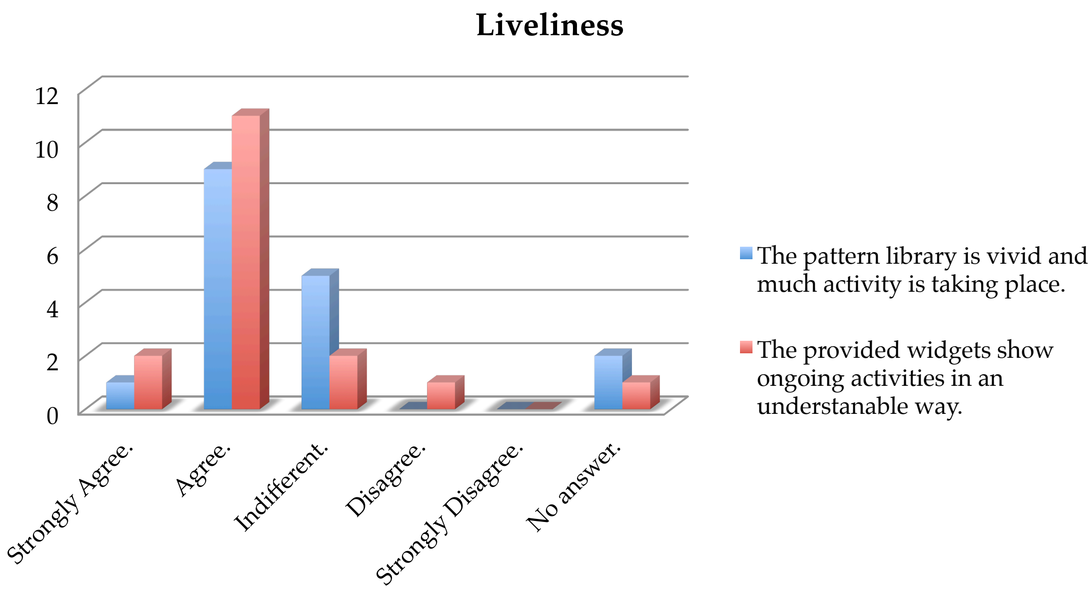


Figure 6.24: Impressions on the pattern library's liveliness and ongoing activities.

Estimation on concept potential

Finally, the survey covered the estimation of the future potential of the EPL approach that is illustrated in Figure 6.25. After the practical usage of the prototype, the participating teams were asked for their opinion about the future extensibility of the approach with regard to further improvements and project-related adaptations. 10 teams agreed that the approach appears as extensible for future development, 3 teams strongly agreed to the statement. 1 team was indifferent 1 disagreed. 2 teams did not provide an answer. Concerning agreement, the potential of the integration of the EPL concept into project workflows was estimated the same way as the preceding question. Free comments on these topics revealed that chances for the approach were seen to use the portal as underlying knowledge structure for study courses.

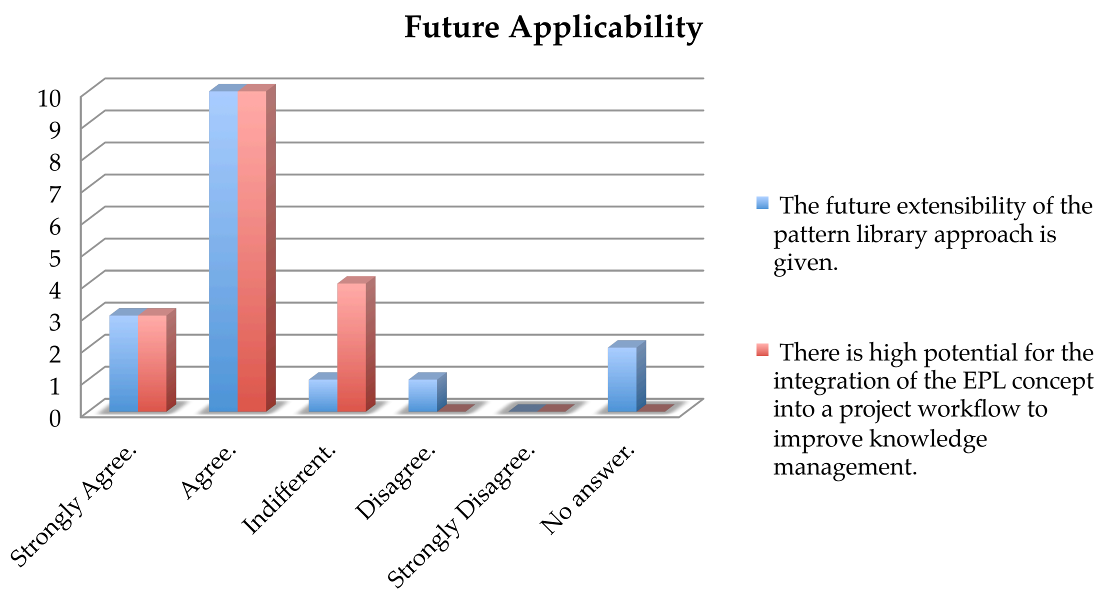


Figure 6.25: Estimations on future usage of the EPL approach and its potential embedment into project workflows.

Results support the approach

In summary, the results of the workshop study showed that the EPL concepts regarding the process, role model, hierarchy structure and maturity concept were understood and applicable. A supporting platform needs to remain easy to use with regard to navigation and submission. The existing prototype mainly fulfills this requirement. Design and interaction refinements should be performed in close cooperation with users in a continuous project context. The approach and its technical realization are on the right track. A thorough introduction to the concepts seems to be supportive although the participants knew about the pattern concept. Still, in the EPL contexts the perspective, processes, roles and rules should be explained before usage. Explanations should be given on demand for quick recovery. Based on the promising feedback of the study, smaller refinements on the widgets and presentations were made to the prototype before preparing it for a remotely executed workshop as described in the next chapter.

6.4 Conclusion

The realization of the abstract EPL concept was performed in two steps. First, a lightweight pattern library prototype was published and provided to the members of the BRIDGE project in order to better communicate the envisioned approach. This allowed productive discussions based on a concrete system. An implemented basic set of requirements is discussed. This first version of the prototype was used for practically explaining the evolving pattern library concept and start collecting patterns in the project. Practical work with the system helped to identify further needs and chances for improvement. Initial prototype

Based on the results of the usage period, a refined version with advanced functionalities for the contribution of patterns, feedback, ratings and evidence was created in a second step. It incorporates the developed role model with its responsibilities and rights together with improved means of visualizing the library structure as a directed acyclic graph. Quality gates for pattern formulation and validation were implemented based on the maturity states and transition rules described in the abstract EPL concept. The description of two iterations of the realized evolving pattern library prototype fortifies the potential of the application of the EPL approach in distributed research projects. For demonstrative purposes, concrete configuration parameters and system screenshots were taken from the version prepared for the final validation in the scope of the project BRIDGE. The implementation allows for ad-hoc reconfiguration of the quality gates and full control of all contributions. Final prototype

An acceptance study of the refined implementation was conducted at the ARL Summer School event at Reutlingen University in June 2013. The look and feel as well as the applied rules for the quality gates were adapted for the workshop for which patterns from the domain of service-oriented architectures and cloud computing infrastructures were used. Reconfiguration took into account a smaller amount of patterns and hierarchy levels. The didactic aim was the dissemination of the pattern concept together with domain knowledge and, at the same time, the review of existing patterns by the workshop participants. Although the workshop agenda only allowed a time-lapsed presentation of a real project setting, the thematic connection to the project work was established. Chances for improvements were identified from the post-workshop survey and taken into account for future work. Acceptance study

With the general acceptance of the technical implementation, the next chapter describes the application of the prototype in the project BRIDGE in a remote pattern collection workshop spanning several weeks. The focus lies on the showing the feasibility of the EPL approach in an existing research project environment. Final validation

Chapter 7

Final Validation in the BRIDGE Joint Research Project

Pattern formulation in parallel to project work

This chapter presents the practical feasibility study of the evolving pattern library concept in an existing research project setting. In the scope of the project BRIDGE, members from different research institutions and industry partners with different backgrounds concerning expertise, research methods and processes cooperate and conduct research in the domain of large-scale emergency response scenarios. During the design of the first iteration of the EPL prototype and the pattern collection workshop held in January 2012, the BRIDGE pattern library could be seeded with satisfying set of patterns and pattern ideas. The final remote pattern collection workshop was conducted in the scope of BRIDGE in July 2013 and made use of the refined EPL prototype that was described in the last chapter. With the help of the refined prototype, the EPL concept could be fully communicated to the project members and applied in parallel to the project work throughout a remotely distributed pattern collection workshop. The validation shows the feasibility of the EPL concept via a technical implementation.

Participants and workshop agenda

Section 7.1 describes the composition of the participating research and development personnel together with the workshop agenda. Different assignments were published during the workshop period that should be completed by each participant when time was available. No deadlines were given and contributions could be made at any time in an asynchronous manner. Meetings and appointments were avoided to reflect the distributed and asynchronous working efforts of the project partners. The means for showing activity of the members of the pattern library played an important role for informing the participants about latest contributions and pattern developments.

Usage within the BRIDGE project

The conducted collaborative workshop increased the number of existing patterns and helped refining them. Section 7.2 provides an overview of the patterns library structure in September 2013. New patterns in addition to the seeding set at the start of the workshop were formulated and mutually reviewed. All existing and newly submitted patterns were set to the maturity state *under consideration*. Due to reviews and comments, formulations were modified. Based on user ratings and the provision of evidence throughout the usage period, the pattern maturation process could be pursued.

Validation via survey

At the end of the validation period, the participants filled in an extensive questionnaire that is included in Appendix D. Besides the successful integration of the approach into the project work, the survey was intended to collect more information on special technical and conceptual topics and to identify rooms for improvements and learn about acceptance and understandability of specific features. Section 7.3 discusses the survey results in detail as well as qualitative feedback as input for future refinements.

7.1 The BRIDGE Distributed Pattern Collection Workshop

Distributed work

The EPL approach intends to encourage project members to contribute domain-related knowledge as open problems, solutions contemporary research results into an evolving pattern library repository. Contributing as well as working with existing content happen on an asynchronous basis since the different work package members are often spatially distributed and access the pattern library at different times. Agreements on common meetings for providing or consuming pieces of knowledge should be avoided such that the pattern library is used when it fits into the daily working schedule. Further discussion which is important within a research context still happens personally during meetings or phone conferences in parallel.

Group composition

Following the described intention a distributed pattern collection and usage workshop was organized in July 2013 after the relaunch of the BRIDGE Pattern Library platform[1]. The new version integrates the advanced functional and visualization-related features as described in Section 6.2.2. In total, 19 participants working in 7 different institutions that were spread over 7 different countries could be obtained for the workshop. The participants' main efforts for the BRIDGE project were conducted in the work packages concerned with domain analysis, interaction and system design, software architecture, validation, exploitation as well as socio-ethical and legal aspects. The number of project members of the BRIDGE project that take part in the project on a regular basis is amounted to 60. Thus, a participation quota of over 28% was reached by the survey. The distribution of institutions, countries and work packages is considered as representative for the BRIDGE project.

Project-external participants

In addition to the regular project members, 2 guests from the domains of service-oriented architectures and flight planning took part in the open collaboration process and the survey. Both guests provided ratings and comments. One guest additionally contributed a pattern regarding communication failures in emergency situations. The contribution was motivated by already existing process-related patterns.

[1] http://pattern-library.sec-bridge.eu

Professional background

From the workshop participants, 16 filled in the online questionnaire that is shown in Appendix D. The members of the survey group were between 29 and 46 years old and had an academic background reaching from student status (1) over Bachelor's (1), Master's degree (11) to PhD (3). Their professional backgrounds were split up into IT-related (10) occupations, interaction and interface design (3), physics (1), social sciences (1) and flight business (1). The participants indicated their main activities in their primary projects as damage containment, domain analysis, interaction design, networking and security, requirements engineering, programming and implementation, project management, research in software architectures and software design. In addition to their primary tasks within their organization, they stated fieldwork analysis, linking empiric findings to designs, scientific publications, collaboration and coordination, activity leads, requirements engineering, programming, organization, GUI prototyping and implementation, application concept design, research planning and design, administrative tasks, field observations, user testing and software engineering. 10 participants said that their work was primarily related to research projects, 4 saw their work connected to research and industry and 1 participant responded to exclusively work in industry projects. 1 participant did not provide an answer.

Pattern and ER domain knowledge

Asked for their knowledge level regarding patterns and the domain of emergency response, only 1 participant for each question answered to have no experience in these domains at all. For the others, rough to good knowledge was at least present as shown in Figure 7.1. It reveals that all participants have expertise in software engineering. Most of the participants ranked their knowledge as expert and good. A similar distribution can be seen for the level of knowledge for user-centered design in which the majority consisted of experts. Comparing the results to their project activities and professional backgrounds, these estimations match the education and activity descriptions.

Expertise with pattern

Answering on their level of knowledge regarding patterns and the frequency of using them, 2 participants stated not to make use of design patterns at all. 1 did not provide an answer. The rest of the group declared to use them at least once a year (rarely, 3), once per half year (sometimes, 4), at least once per two months (often, 2). 4 participants made use of design patterns on a regular basis, i.e., at least once per month. Patterns were used for different purposes such as software and application development, domain analysis, references to custom design, web design, prototyping and as start for research in the field of cloud computing.

Pattern usage

Regarding the formulation of patterns, 7 participants have not written patterns at all, 2 did not provide an answer and 4 declared to sometimes formulate a pattern on a half year basis. 1 Participant answered to formulated patterns at least once per two months. The remaining 2 answers stated that at least once a year patterns were written.

Patterns in daily work

The distributions reveal that the group consists of people who primarily make use of patterns in a consuming and learning manner such as the original pattern approach suggests. Looking at the influence of pattern on their daily work, the results are almost balanced. The group was not used to formulate patterns on a regular basis.

Figure 7.1: The participants' expertise with patterns, in the domain of emergency response, software engineering and user-centered design.

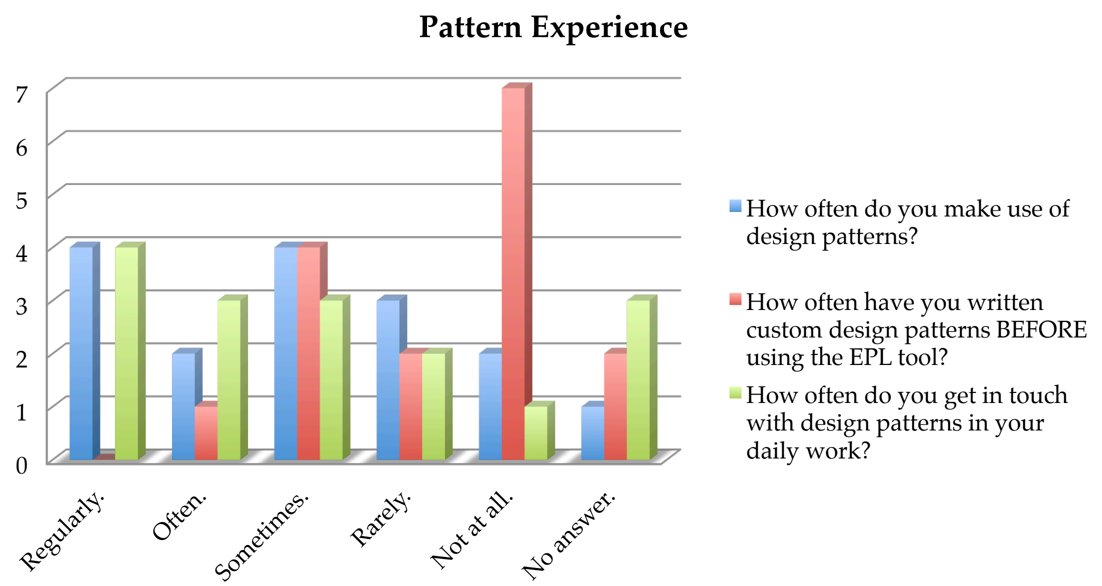


Figure 7.2: The participants' experience with patterns and pattern formulations.

7.1.1 Workshop Agenda

Cumulative tasks

Since the workshop took place in a decentralized setting without scheduled meetings, a workshop agenda was created that spanned three time periods. In each period, a predefined task should be completed that took no longer than 30 minutes. This way, a total interaction time with the refined pattern library of 90 minutes was targeted and served as guidance for the participants what to do next. Since not all participants were available on the same week, the workshop agenda was repeated twice in subsequent weeks. Throughout the whole interaction period of three weeks, all participants could interact at their own pace with the prototype, besides the suggested agenda. Especially in the third week, some participants used the opportunity to submit more ratings and comments as reviews for existing patterns.

Agenda and concept communication

Before starting the first session, a brief change log of the newly developed features was sent to the participants. A remote live introduction was offered that was attended by one third of the participants. Additionally, the start page of the pattern library presented a summary of the maturity process, the role model and the workshop agenda (cf. Figure 6.13). For each assignment, a description was sent out to the participants on the day it started and an announcement was made via the newsticker widget. With the start of each new task, the preceding one remained active such that activities were cumulated. In principle, every activity could be performed at all the time. The purpose of the agenda was to stepwise introduce the different features of the concept implementation. The pattern maturation process and its intention were summarized together with the defined maturity states. Rules and thresholds for quality gates regarding a pattern's formulation quality as well as its validation were communicated.

Pattern contribution

For the first assignment, each participant should extract two promising pattern candidates out of his current working field and to formulate it in the pattern library. In case more patterns were found or additional problems that are currently tackled but still need a solution, the participants were encouraged to provide those as well. After the end of that period, the *Submission Control Board* consisting of the author of this thesis and one pattern expert from the interaction and system design work package assessed the submissions and put them online after the first quality check.

Foreign pattern rating

The second assignment that was sent out one day after the first one, requested the participants to read through the newly structured pattern library and review arbitrary patterns. At least, four patterns should be rated. Optionally, comments regarding formulations could be given and evidence could be assigned. The participants were asked to pay attention to the widget indicating patterns that still need ratings.

Evidence

The final assignment regarding the participants' interaction with the refined EPL prototype was to assign evidence for the patterns they submitted. Again, at least two own patterns should be supported by evidence. Two foreign patterns that were still in the state *under consideration* should be spotted and supported or refuted by evidence.

7.1.2 Pattern Library Setup

Initial seeding of the library

The EPL platform was seeded within the existing patterns that were formulated within the collection workshops using the first prototype (cf. Section 6.1). This way, an initial population of the library with 28 patterns served as starting point for the first pattern review assignment of the remote workshop. While reading the existing patterns, there was a chance for new and alternative thoughts and points of inserting current findings from the individual project work. All patterns were in the maturity level *under consideration* since they at least contained information about the context and treated problem as well as a suggested solution.

Hierarchy Levels

All patterns were arranged within the hierarchy levels that were proposed by the concept adaptation in Table 6.2 in Section 6.2.1. Associations between patterns that refine ideas of upper layers were proposed as examples for pattern associations. All preparations of the prototype were subject to collaborative discussion. It was communicated to the participants that no assignment was to be considered as fixed. Recommendations for improvements on all aspects were welcomed and encouraged.

Rule parameters

The rules were set similarly to the ARL Summer School prototype. Because of the concentrated validation phase of three weeks and a representative group of more than 25% of the whole BRIDGE project's active member size, the maturation steps should be triggered for at least 75% of the formulated patterns. Therefore, the minimal number of ratings was set to 2 and the formulation quality threshold for the rating aspects about readability, understandability and appropriateness was set to 3. For the validation threshold, the PMI was set to 3. Each evidence factor scored $+1$ point for supporting and -1 point for refuting evidence. Thus, for the practical, short time workshop, the more advanced concept of hierarchy level pattern dependent maturity calculations as given in Definition 5.4 was simplified to create an easier starting point for the pattern formulation and review process. The advanced concept was considered as useful for long-term application of a pattern library instantiation.

Evidence factors

A predefined set of evidence factors was presented to the participants with the intention to stimulate thoughts where evidence from the project work, attended conferences or read publications could be found. The initial set of evidence factors consisted of the entries described in Table 6.3 presented in Section 6.2.

7.2 Summary of Workshop Contributions

BRIDGE pattern library as third contribution

The pattern library represents the third contribution of this thesis and shows that the EPL approach can successfully be applied in a real project setting. The presented patterns are beneficial products of the research agenda leading from the requirements analysis to the presented concept of an evolving pattern library. They reflect the current state of discussion within BRIDGE. Therefore, some formulations are briefer or at less mature degrees than others. Many times, ongoing discussions show that reviews by the community are performed that the original author can use for the refinement of the pattern formulation. Thus, the formulated patterns represent the current state of the collaborative knowledge gathering and formulation process. Contributions were made by members working in the fields of domain analysis, interaction and application design, validation and demonstration, legal and ethical aspects as well as dissemination. The author of this thesis took the opportunity to insert patterns based on his own project experience and to trigger discussion on the topics addressed. The whole pattern library structure is online available and listed in Appendix E in a condensed version.

Library state

Throughout the remote pattern collection workshop, the total number of formulated patterns grew to 43. The structure of the pattern library in September 2013 is shown in Figures 7.3 and 7.4. Most contributions were made to the hierarchy levels considering general processes, application concepts, user interface and interaction design as well as technology and system design. Domain practices and laws and ethical considerations are not yet addressed as often as other. All new pattern submissions directly passed the initial quality check such that they were released to the collaborative review process immediately. No pattern was formulated as *open problem*. 24 patterns are in the state of patterns *under consideration*. Given more time and project member participation, the potential for the provision of additional ratings and reviews for these proto-patterns would certainly increase. 17 patterns reached the state of *pattern candidates*. Although in total 31 evidence factors were assigned to patterns, only 2 patterns were approved according to the rules defined for this prototype according to which at least 3 supporting evidence factors need to be assigned.

Project activities reflected in structure

Current activities of the project's work packages at the time the workshop took place are reflected by the pattern distribution. Most participants were concerned with technical developments as well implementations of application and interaction design concepts. Domain knowledge was already fed into the designs such that the patterns were documented in retrospective. The visualization indicates potential gaps regarding domain knowledge which is present for the specialized work package members but not yet distributed within the whole project.

Rating and comments

The discussion function in addition to the quickly to perform rating was used elaborately. In total, 48 comments were given on 30 different patterns. According to the amount of given ratings, 86% of the library contents were reviewed at least once. Especially the activities on patterns in the state *under consideration* encourage the use of a collaborative pattern formulation platform and show that critical dispute took place. In the following, three examples of the BRIDGE pattern library are presented for each pattern maturity state. They exemplary show the different foci of the contributions, reviews made by members and the evidence-supported validation process within the evolving pattern library concept.

A pattern "under consideration"

The first example shows the pattern "Active Influence on Logging" that was formulated for the hierarchy level "Laws and Ethical Considerations". It shows that all required

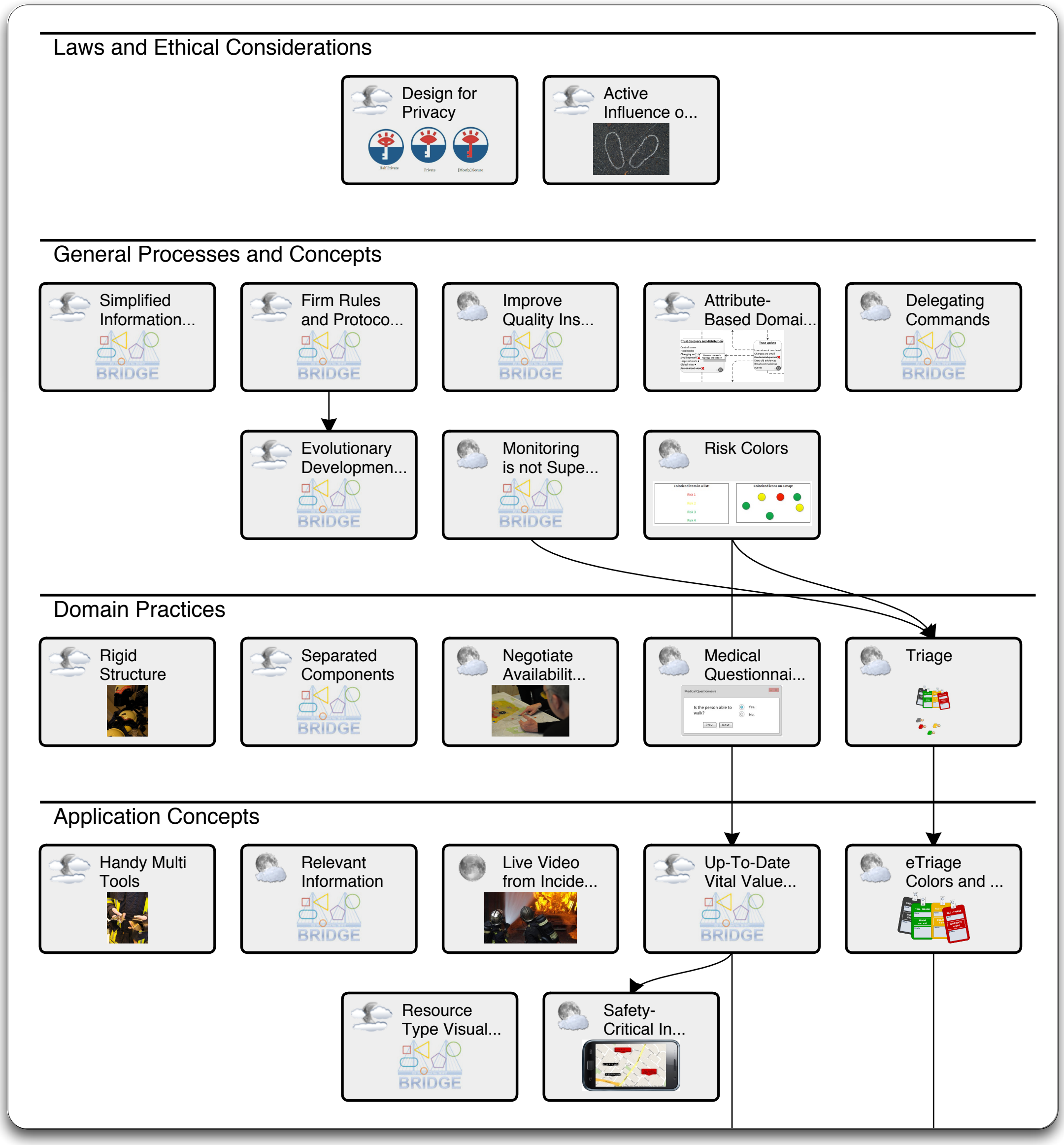


Figure 7.3: The BRIDGE Pattern Library structure in September 2013 - part 1 of 2.

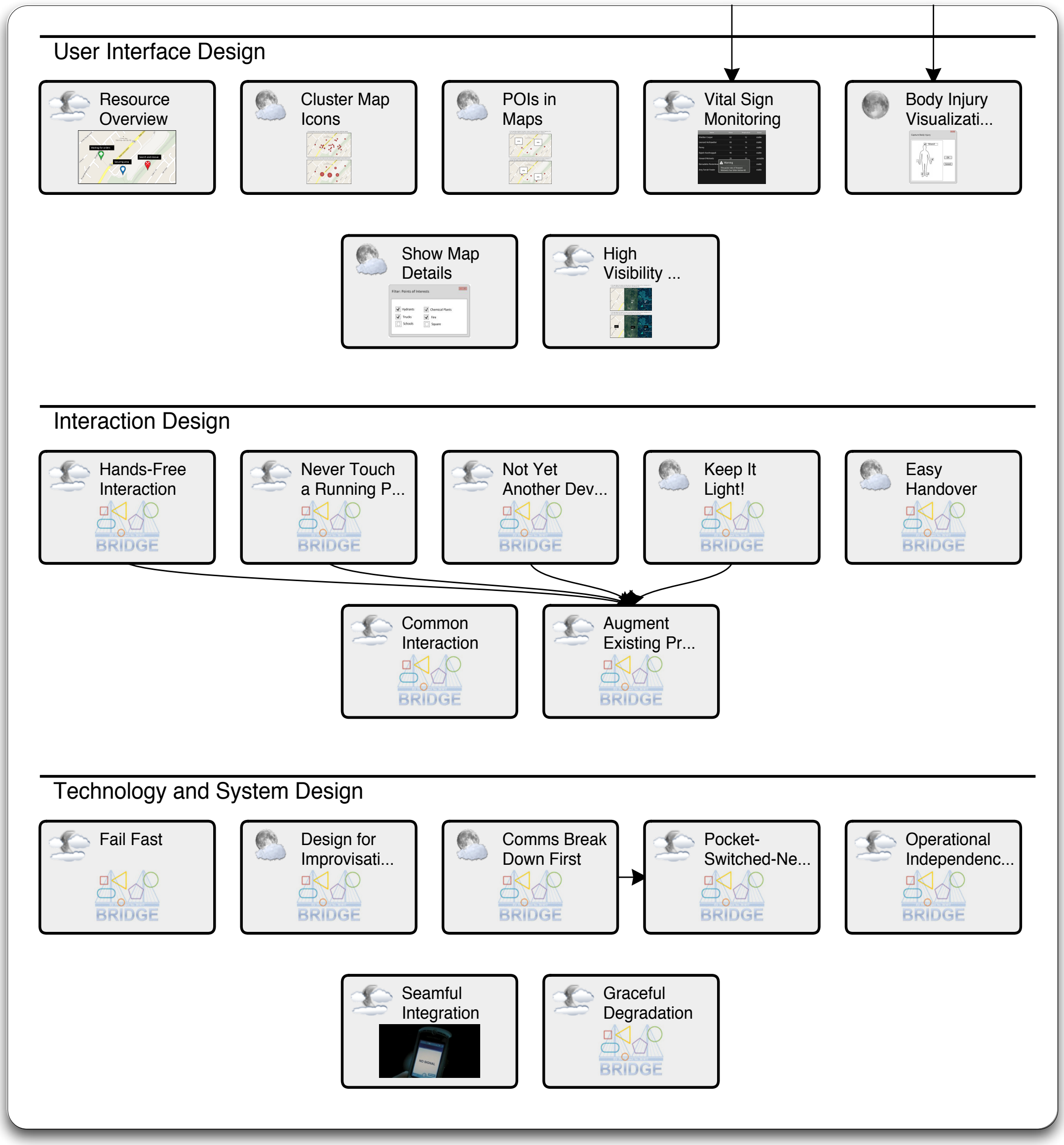


Figure 7.4: The BRIDGE Pattern Library structure in September 2013 - part 2 of 2.

fields are filled and the context of the pattern is described. The problem is elaborated and a solution is presented. According to the pattern maturation process, other readers first have to review the formulations and provide ratings on the aspects regarding readability, understandability and appropriateness. The online version of the pattern would show that one rating was already provided. It is only shown to the user after a custom rating is given. Comment fields allow for discussing the current formulations. From the given comments, it can be seen that structural remarks regarding the pattern name and its problem description were given. Other users focus on the content of the pattern and try to improve the solution or create more patterns from the current one. In turn, the original author can modify his formulations or provide answers to comments. This way, discussion is established. Other users have the chance of deriving similar patterns with differences to this one as proposed by one comment. The pattern example not only represents a starting point for the discussion for a special topic, it also contains current considerations within the project and serves as "food for thought".

Active Influence on Logging

Under Consideration
Pattern Origin: *Adapted to Project*

Context

Emergency responders already log key decisions and events. Technology is part of a transformation of logging capacity and practices.

Problem Summary

Technologies can log human activities and communications that are entered into systems directly or indirectly. However, technologically augmented logging is also part of a transformation of what logging means and what it is used for.

Problem Details and Forces

Logging can be great as it can save vital time, it can take work off people's shoulders, logs can be locked so no-one can change them, machines don't get tired logging, extensively logged real world events can be used for training. It can be used to apportion blame and punishment in retrospect, which, in turn can affect the way in which people carry out their work in the knowledge that everything will be logged.

Solution Summary

There are no 'solutions' - there are lots of ways of managing both positive and negative consequences. For example by installing forgetting, by logging things without identifying persons, by allowing people to turn logging off or erasing their logs.

Comments

- *I would like to suggest some smaller changes: 1. Maybe, the title could be shortened to something like: "Log or Don't Log" or "Human-Managed Log" since otherwise, the pattern name itself is a little too long to be remembered easily.*
  2. The solution part says that there are no solutions. However, I think that the examples presented are solutions that could be applied. Maybe they could be listed or a little further described in the section about solution details and consequences when applying them.
  3. The Problem Summary could be split up into the central message. The rest of the section could be moved to the problem details. This would improve readability and support a first quick look through the pattern before reading on.
  What does the author think about that? :)
- *I understand this pattern that there are pros (technology can log very efficiently / "it can take work off people's shoulders" / etc.) and cons (logging can effect the way people work if they know everything they do is logged) of logging. So the pattern focuses tradeoffs between logging and no logging. What I want to mention is: 1.) I agree that logging has direct impact to technology and systems design. Nevertheless I think that the concept of logging itself should be a pattern on a higher layer. I am not familiar with law and regulations in the field of emergency response but I will give an example from the domain of ERP systems: Logging on business transaction level is regulated by law because a ERP system has to be revision-safe. So perhaps there are also such forces in the field of emergency response that directly lead to a "Logging Pattern" in the layer "Processes and Concepts" or "Domain Practices" which is derived from a pattern in the "Laws and Regulations". Then the "Log-don't log" pattern could be logically derived from those patterns and would therefore be much better placed in the whole domain context of emergency response.*
  2.) Also if there are no such higher level concepts and regulations in the domain of emergency response like mentioned in 1 I still can extract more than one pattern from the problem description: a) logging can be done quite efficiently by technology, so design applications for logging and take off that work from human shoulders b) there is a need for logging at all c) we can extract data for training purposes using some logging mechanisms
  3.) As I can use some sentences from the problem description very easily to formulate solutions I wonder if the problem section isn't describing solutions but rather than problem(s).
  Nevertheless I think this pattern opens up a couple of new pattern ideas so this pattern repo is quite the best place to formulate those ideas.
- *Isn't this necessary like for example black boxes on planes?*

. . . .

Example for a "Pattern Candidate"

The following example shows a *pattern candidate* from the hierarchy level "Domain Practices". The pattern named "Negotiate Availability" has passed the formulation quality gateway. It can be seen that some changes to the formulations of the context were made according to the first comments. Other suggestions were not completely followed. Further comments still provide more information for the improvement of the pattern formulation and inspiration for formulating related ones. The third comment suggest to associate the current pattern to others related in the hierarchy levels that consider user interface and interaction design. Besides the ongoing discussion, formulation adjustments and derivation of new patterns based on given suggestions, members now need to find evidence for the pattern in order to strengthen its validity. One citation from a user workshop is already assigned.

Negotiate Availability

Pattern Candidate
Pattern Origin: *Adapted to Project*

Context

Incident commanders need to coordinate human and material resources. Especially for human resources, the exact state and occupation may be different from the last status update. It is also possible that the current activity covers more than reported.

Problem Summary

Human 'resources' may look available on a screen, but they may not be available in the real situation. If resources that look 'available' are deployed when they are not, in actual fact, available, problems will occur.

Solution Summary

'Deployment' or 'resource allocation' should in most cases involve personal communication.

Supporting Evidence

- *User Workshop: Stavanger Demo II.*
  Description: Summary of a user statement during the prototyping session: If you"re dispatching by dragging that ambulance onto that incident you take away the negotiation. People can't necessarily do that job or not right now or not be the best for it, regardless what is said by the system.

Comments

- *1.) Shouldn't the context of the pattern also mention that a precondition of this pattern is that "availability" or more general "status" of human resources have to be captured some way and also displayed on a screen!?*

2.) I think the part "...but they may not be available in the real situation." in the context section is more a description of the problem rather than a description of the context.
3.) I think the meaning of the word negotiation in the title goes far beyond just communication as it is mentioned in the solution summary. I will give an example from my own experience from the time I was working as an emergency medical assistant: There is a special kind of radio transceiver in every ambulance. The staff of the car can send their current status to the rescue coordination centre just by pressing a button on that radio. Pressing 1 means the status of the ambulance is "available", once the rescue coordination centre deployed a mission to the ambulance the staff can signal that they are on the way to the place of action pressing 3, pressing 4 they signal we arrived, 7 means we got the patient and drive to the hospital and finally when the patient is at the hospital the staff makes the ambulance ready again and then communicates the status "available" pressing 1 again. What I want to show with this example is just that there is not just communication but communication following some rules. Especially when the rescue coordination centre deploys a job to an ambulance with status "available" the staff of this ambulance has to acknowledge the job. So that is what I want to mention as negotiation, not just communication. Okay, so what can we derive for the pattern: a) the concept of communication should be elaborated in the solution sections to become much clearer and to match with negotiation in the title. b) in my opinion the concept of negotiation leads directly to some kind of acknowledgement of deployments.
4.) The mentioned findings lead directly to some requirements and therefore patterns in the layer "User Interface Design" and "Interaction Design". Especially I see a connection between the pattern "Common Interaction" and this pattern since Common Interaction describes "Enable them to consuming information but also to collaboratively provide and interpret the information.". Referring this sentence to 3 I feel that we can apply this pattern to achieve "Common Interaction" at least partly.
5.) The solution illustration should show the negotiation aspect, when I look at the illustration I think of "planing" and "decision making" but I can't see communication and negotiation.

- *Addition to 4.) I also think that there is a new pattern in the layer of user interface design which describes defined negotiation possibilities using buttons ore something else to establish the functionality of common interaction and "Availability is a matter of negotiation".*

....

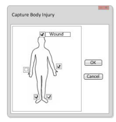
The *approved pattern* describing the concept of "Body Injury Validation is placed in the hierarchy level "User Interface Design". It is built on other patterns that deal with monitoring issues and the domain concept of "Risk Colors" that need to be understood for

Example for an "Approved Pattern"

the ”Triage” and ”eTriage Colors and Icons” patterns. The practices and concepts explained in these hierarchy levels relate to the current pattern as visible from the pattern library structure shown in Figures 7.3 and 7.4. According to the rules and thresholds defined for the remote pattern collection workshop, the rating threshold was passed and enough evidence items were collected to consider the concepts and solutions described by the pattern as valid and certainly reusable. Three supporting evidence items from research publications are cited that were accepted by the *Submission Control Board* and therewith regarded as counting evidence. Still, the pattern is subject to discussion as visible from the comments. Suggestions for improvements on the formulations and relations to other patterns as well as supporting comments are given.

Body Injury Visualization

Approved Pattern
Pattern Origin: *Derived from Project*



Context

During search and rescue operations, a victim's injuries need to be documented and transmitted to the hospital for further medical treatment.

Problem Summary

Paramedics need to quickly assess the injuries of a victim during the triage process. The documentation is used by follow-up medical personnel.

Solution Summary

Therefore, show illustrations of the human body according to which injuries can be indicated.

Supporting Evidence

- *Research Publication: Designing an Emergency Medical Information System for the Early Stages of Disasters in Developing Countries.*
  Description: (Sutiono et al., 2010): Sutiono, A.; Qiantori, A.; Prasetio, S.; Santoso, H.; Suwa, H.; Ohta, T.; Hasan, T. andMurni, T. - The Human Interface Advantage, Simplicity and Efficiency Journal of Medical Systems, Springer Netherlands, 2010, 34, 667-675.
- *Research Publication: ARTEMIS: A Vision for Remote Triage and Emergency Management Information Integration. Dartmouth University. Nov. 2003.*
  Description: S. McGrath, E. Grigg, S. Wendelken, G. Blike, M. De Rosa, A. Fiske, and R. Gray
- *Research Publication: Electronic Triage Tag and Opportunistic Networks in Disasters.*
  Description: Martin-Campillo, Yoneki, and Crowcroft - ACM Special Workshop on the Internet and Disasters in CoNext, 2011.

Comments

- *The text is a bit shallow, e.g. it is no clear who the "user" is, who should capture the injuries. That could be a problem if a non-professional should capture injuries because he just doesn't have the knowledge. On the other hand, a professional might loose important time with capturing injuries. That said, I think the pattern could need some supporting evidence, a problem description and for sure some illustrations.*
- *This item should obviously be related to other triage patterns*
- *This is a very clear description.*

. . . .

Proof of concept

The formulated and reviewed patterns are a beneficial outcome of the application of the EPL concept in a research project setting. The current state of the pattern library represents a feasibility study of the EPL approach and encourages the further the implemented collaborative platform in a research project setting. Since validated patterns capture reliable solutions to problems it would be too early to present the mined patterns as approved. More evidence needs to be found in order to fortify the proposals made. Decreasing the minimum number of required evidence factors only to enforce the advancement to the highest maturity state would not be in the interest of the overall pattern concept. Continuous research within the project allows the pattern library to grow.

Post-study survey

In order to abstract the findings and learn about the general acceptance and applicability of the presented approach, a post-workshop survey was conducted that asked about concrete details of the approach regarding the pattern formulation and maturation process, visualization support, interaction and understandability. The next section discusses the individual questions and results in detail.

7.3 Post-Workshop Survey Results

Survey structure

Interacting and working with the system was an elementary step to actively involve the research and development personnel such that they can give statements about their experience. After the usage period within the distributed pattern collection workshop, participants filled in an online questionnaire as shown in Appendix D. It was subdivided into an introductory part considering the participants' backgrounds as presented in the preceding sections and topics related to the EPL role model, the rating, the evidence model and implemented assignment functionality as well as pattern maturity model and the used metaphor. Further topics covered the hierarchy model, the overall perception of the pattern library's liveliness in conjunction with the implemented widgets and visualization concepts. Finally, general estimations about the understandability and usefulness of the EPL approach as well as its implemented should be given.

7.3.1 Role Model and Rating Functionality

Role model and submission check

After gathering participant-related information, the first block of questions considered the understandability of the presented role model and the concept of advisory board of the EPL approach. After acting in the role of *Members* and *Authors*, the participants were confronted with an excerpt of the decision making process of the advisory boards. They had to wait for their submissions to be accepted. The assessment by the *Submission Advisory Board* was made on a daily basis. All pattern formulations instantly fulfilled the minimal requirements. Figure 7.5 shows the diagram summarizing the answers.

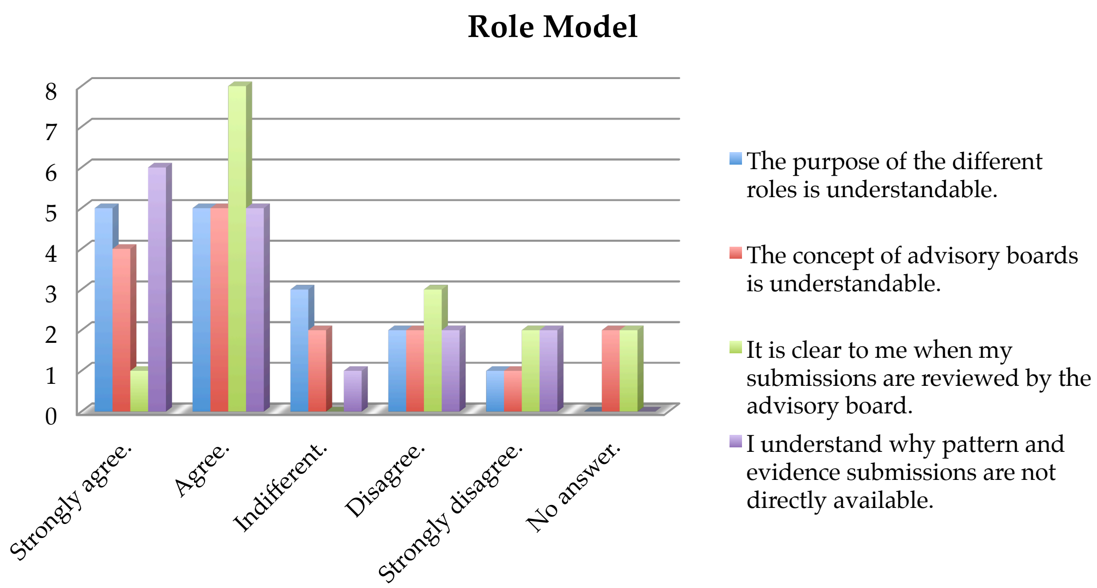


Figure 7.5: Results regarding the understandability of the different roles and advisory boards.

General understanding

Almost all participants replied that the role model was understandable except for 3. A similar situation can be seen for the role of the different advisory boards. Besides 3 disagreements, 2 participants left out the answer to this question. Although the model was explained on the front page, an initial briefing, preferably in a group discussion or via a remote presentation, seems useful.

Assessment

The daily assessment of submissions and the necessary waiting time for publications was understood by the majority. 5 disagreements on the implemented mechanisms and 2 not given answers suggest, in conjunction with qualitative comments beside the Likert scale questions, that more information should be given during the submissions process. Additional explanation before and after contributing should be displayed more prominently in order to make sure that as many users as possible understand the ongoing process they trigger with their action. For submitted evidence, the situation is very similar although it seems that there was a general understanding that these kinds of submissions needed review. From the qualitative feedback the conclusion is drawn that many participants expected to directly see their contributions online.

Rating and commenting

Questions dealing with the rating concept as illustrated in Figure 7.6 reflect the opinion that, except for 1 participant who did not provide an answer, all agreed that rating be performed easily. The meaning of the different quality aspects was clear to the majority as well. 2 indifferent opinions, 3 disagreements and 2 abstentions in combination with the comments given on the aspects suggest to more clearly explain the differences or to further investigate what kind of aspects are considered as relevant.

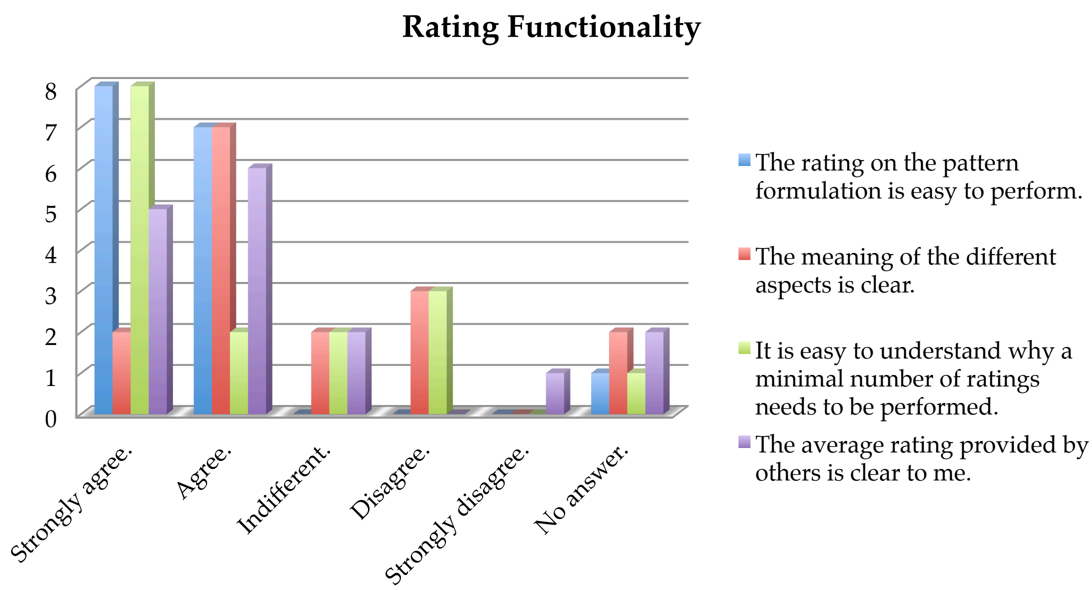


Figure 7.6: Answers on the rating functionality and underlying rules.

More information on rating rules

The need for a minimal amount of votes is shown in the interface that tells the user how many votes are still necessary. However, this does not seem to be enough for all participants. In total, 10 agreements, 2 indifferent opinions, 3 disagreements and 1 answer not given suggest to improve the rule explanation and feedback for users. Contrarily, the concept of the average value that is calculated on all ratings was considered as clear for all participants except for 2 being indifferent, 1 strong disagreement on that question and 2 ones who did not answer the question.

General understanding

Figure 7.7 shows the understanding of the consequences caused by rating a pattern. 8 participants stated that the consequences were clear to them. Again, some more explanation would support understandability as 4 indifferent opinions, 2 disagreements and 2 abstentions indicate. The purpose of the rating function in order to give feedback on the formulation quality for a pattern was mainly seen as easy to understand. 7 and 4 participants strongly agreed and agreed, respectively on this question. Still it seems that some additional information may be needed to better convey this meaning since 1 indifferent opinion, 2 disagreement and 2 not given answers hint on a lack of clarity.

The function to add comments to a pattern was seen as easy to use by 14 participants. 1 voted to be indifferent about this issue and 1 did not provide an answer. Members can change given ratings at any time and only the last one is taken into account. The answers are similarly distributed as in the last question, with a slight shift to indifference. Help texts might be necessary to better explain the way ratings work internally.

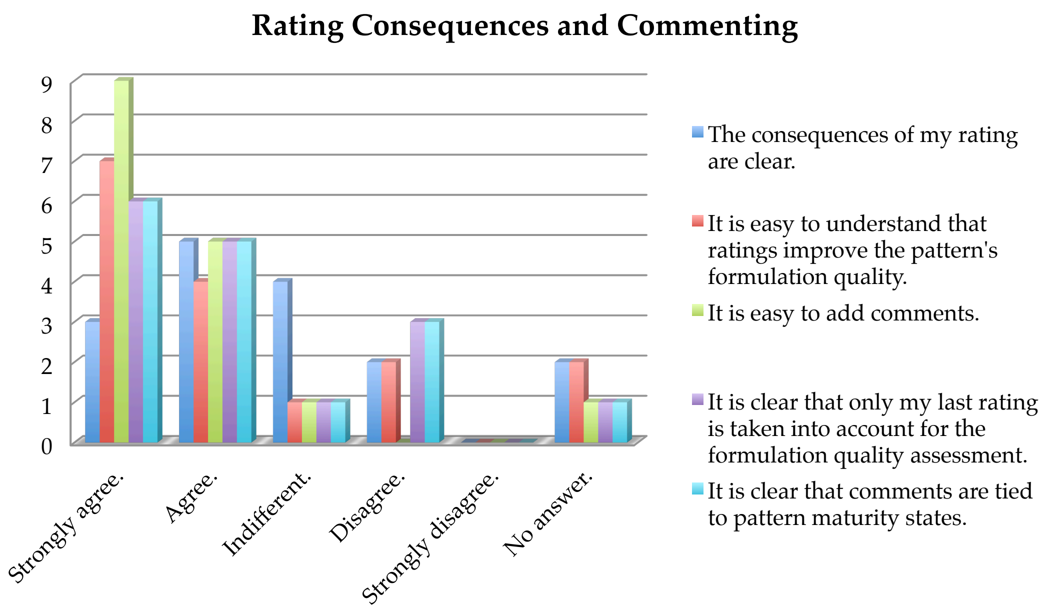


Figure 7.7: Understanding of the consequences of ratings and comments.

Link between comments and states

Although the connection between comments and pattern maturity states was understood by the majority of the participants, the presentation should be made clearer. 1 indifference, 3 disagreements and 1 not given answer indicate that the participants had problems finding given comments or reviewing them after submission. Qualitative feedback in this regard supports this impression. Some participants demanded for better ways to administer given comments. Later changing and moving to another pattern maturity state should be allowed. The link between comment and maturity state should be better indicated.

7.3.2 Contributing, Pattern Maturation Process and Library Structure

Interaction needs improvement

The majority of the participants accepted the overall usage of the pattern library as illustrated by Figure 7.8. Future improvements should focus pattern submission and management. It seems that the corresponding functionality was not found directly. The latter topics had the most not given answers, i.e., 5 and 7, respectively.

Concerns on scalability

Concerns arose about the scalability of the process. 1 participant was uncertain whether the visualization will scale well if the population of patterns increases further. In a well

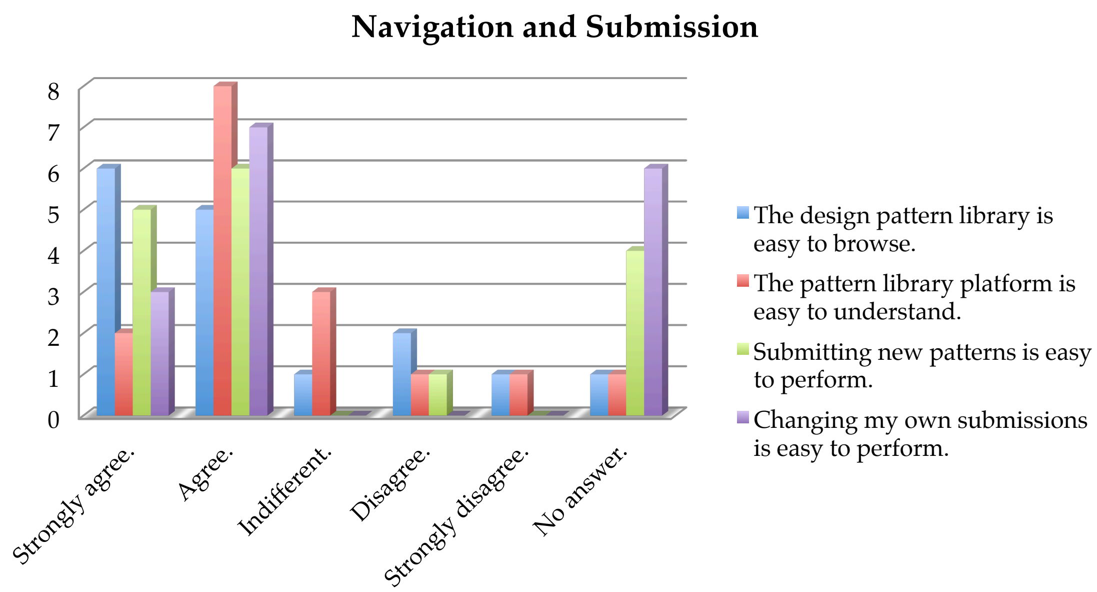


Figure 7.8: Feedback on navigating the library and submitting patterns.

populated pattern repository, an efficient search function besides the graph navigation was considered as necessary. A dedicated search view should also provide means to systematically sort out patterns based on specific attributes, tags and maturity states.

Clarity of the maturity concept

Asking for the comprehension of the pattern maturity concept, the given answers show acceptance by the majority of participants as shown in Figure 7.9. The underlying rules and meaning of the different maturity states were seen as clear in a similar manner. Since the concept presented in this thesis varies from the traditional way of using and formulating patterns, misunderstandings and uncertainties on the users' side seem to have arisen. The pattern library is not always self-explanatory. There is a change that a longer usage period will manifest learning effects such that doubts and uncertainties can be overcome by watching the development of the library and other users' activities.

Need for alternatives metaphors

From this survey, in contrast to the one conducted in the ARL Summer School (cf. Section 6.3), the used metaphor for visualizing the maturity states was not clearly accepted as visible from Figure 7.10. Results on the questions dealing with the understandability of the metaphor and the general appeal of the used icons are spread over the whole spectrum of choices. Although the visual representation of a pattern's maturity was determined during a workshop conducted in the BRIDGE project, it was not really accepted in its present form. More general metaphors like percentage scales or different kinds of traffic lights with special colors were recommended.

Evidence assignment

Figure 7.11 shows the distribution of answers to the questions regarding the evidence assignments. In summary, the concept of evidence factors and the influence on a pattern's maturity state were understood. The same can be said concerning the difference

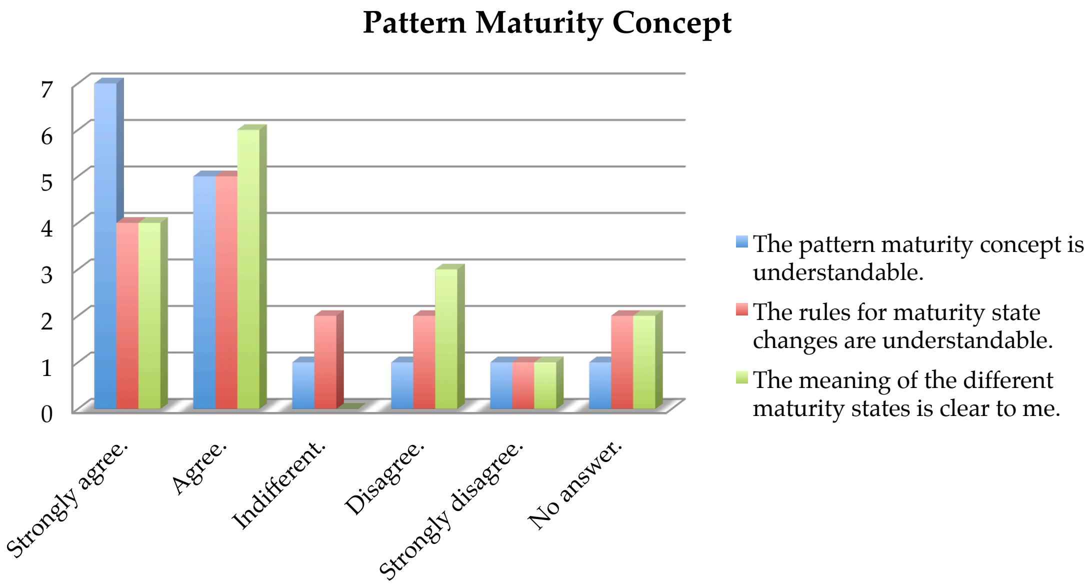


Figure 7.9: The pattern maturity concept was understood by the majority.

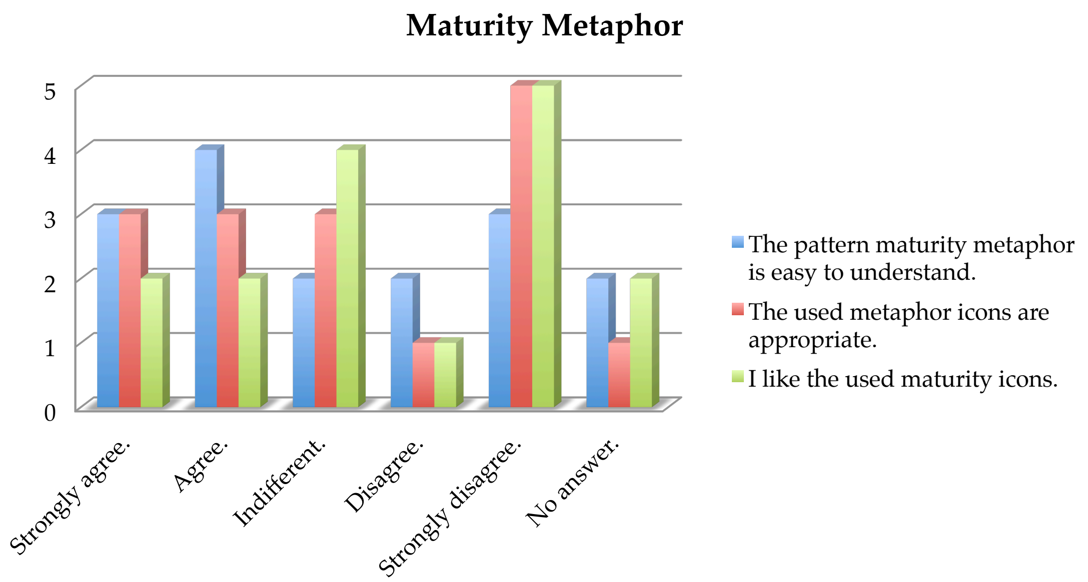


Figure 7.10: The currently implemented maturity metaphor was not universally accepted.

between supporting and refuting evidence. Answers on the evidence assignment show that the majority considered this function as easy to use. Indifferences, disagreements and 3 to 4 answers not given for each of these aspects suggest that the user interface and messaging system should be improved in future iterations. Given comments on this question group support this impression. Few participants found the evidence provision dialog "too academic" and the list was considered as "too long".

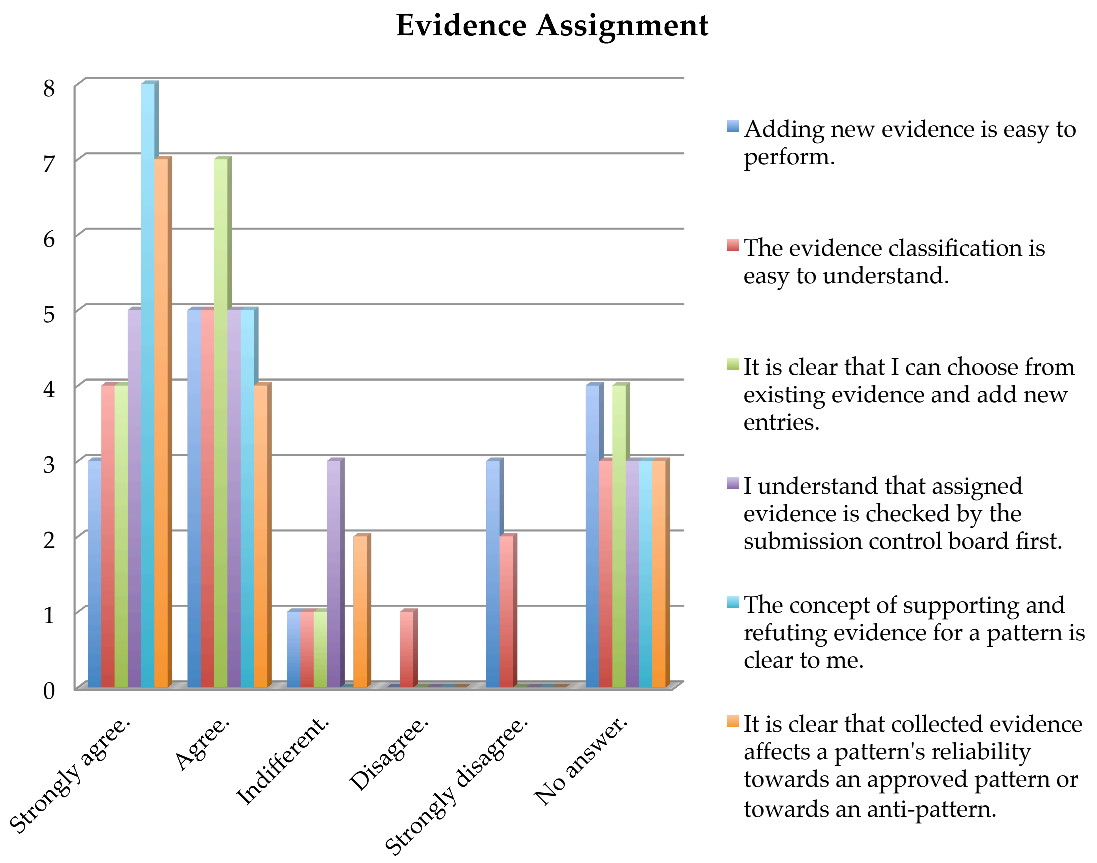


Figure 7.11: Answers on the evidence concept and assignment process.

The need for peer-reviewing submitted evidence items by the *Submission Control Board* was seen by 10 participants. 3 votes stated indifference and 3 answers were not given.The distinction between supporting and refuting evidence was clear to 13 members of the study group. The effect of evidence on a patterns development towards or from an anti-pattern was seen and widely understood. Influence of evidence

In general, the hierarchy concept was considered as understandable and the proposed levels were accepted by the majority of the participants. Figure 7.12 reveals that some authors were uncertain of assigning their contributions to a specific hierarchy level. Future discussions should elaborate on naming, additional levels and the possibility of introducing additional tags. However, the structure proposed at the beginning of the Hierarchy concept workshop was widely accepted. The intention of ordering the hierarchy levels by their level of abstraction was understood.

Additional patterns

One further intention of the hierarchy was to help project participants to spot where additional knowledge should be placed within the pattern library. 7 agreements indicate that the concept is on the right track. 4 abstentions, 3 disagreements and 2 not given answers place a demand for improvement.

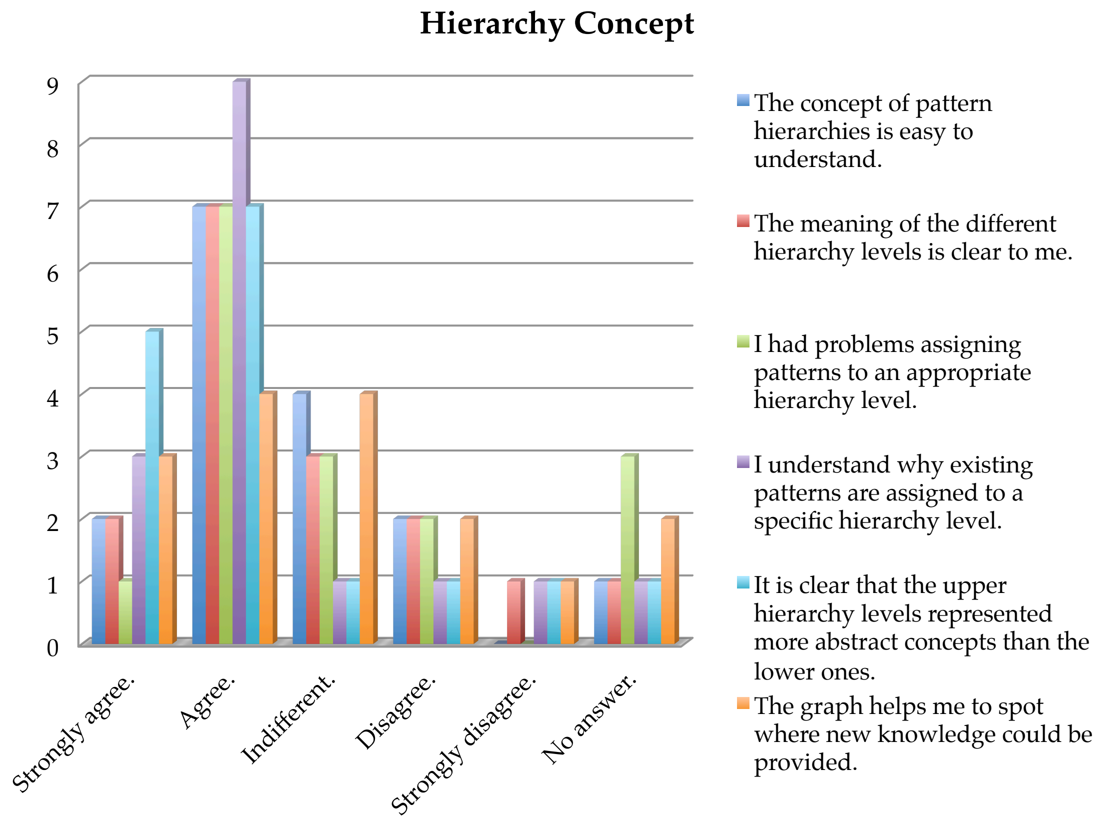


Figure 7.12: Survey result regarding the hierarchy concept.

7.3.3 Information, Liveliness and Motivation

Submission handling

The feedback on the comprehensibility of activities related to made submissions and their current state are shown in Figure 7.13. The number of not given answers, together with the indifferent opinions and disagreements indicate that better explanations and overviews of the authors' submissions should be provided. The dedicated menu entry for custom submission was eventually not perceived or understood. Qualitative feedback did not reveal hints on this issue.

Useful widgets

Feedback on the specialized widgets arranged around the template that focus on differ-

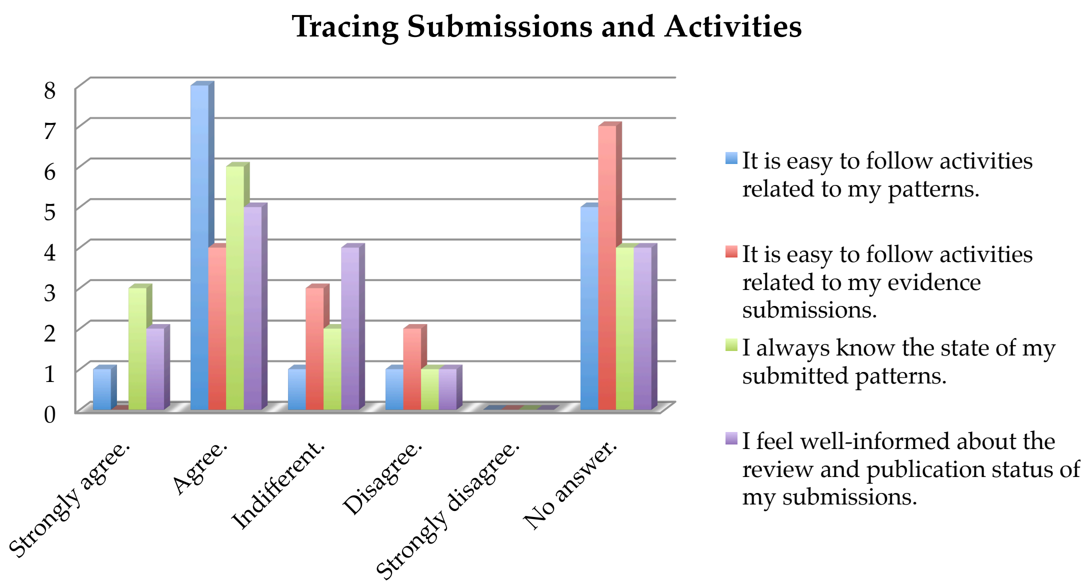


Figure 7.13: Answers regarding the understanding of ongoing activities.

ent aspects of the pattern library is summarized in Figure 7.14. In summary, all of them fulfilled their purpose and were generally accepted. The best acceptance was gained for the newsticker, widgets suggesting patterns that needed more ratings and evidence as well as the hall of fame. Answers show that the latter did not help to motivate users to take action based on others' activities.

Widgets and motivation

As mentioned by the questions about the hall of fame widget, the different ways of preparing information intended to increase the individual motivation of the participants to contribute and review existing patterns. Figure 7.15 shows that the widgets were regarded as important to provide an overview of activities as stated by 2 strong agreement and 10 agreements. The answers were almost equally distributed across all options except for strong disagreements which was not chosen at all. It seems that motivation cannot solely be created by displaying hints on other users' activities or suggestions for next steps. Future refinements should clarify what kinds of motivational factors need to be chosen. Related work by Leacock et al. [2005] and preceding work by Prause et al. [2010a] could serve as a starting point.

Impressions on liveliness

In conjunction with the discussion of different widgets, the participants were asked about their perception of the pattern library's liveliness. The results summarized in Figure 7.16 show a mixture of impressions. 7 participants found that the library was vivid and much activity was taking place. 3 opinions were indifferent about this point and 5 disagreed. The number of not given answers was 1 for all questions asked on the topic of liveliness. Awareness of other users and ongoing processes could be created for most of the participants and relevant events were shown. According to the number of

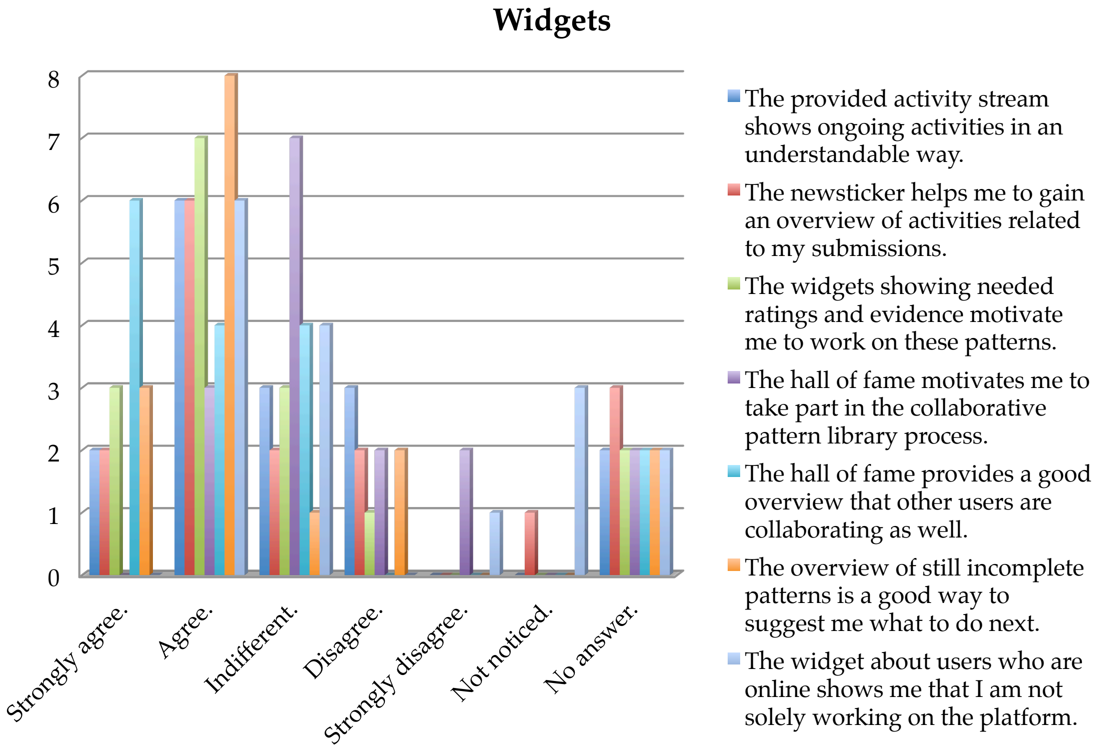


Figure 7.14: Acceptance of implemented widgets.

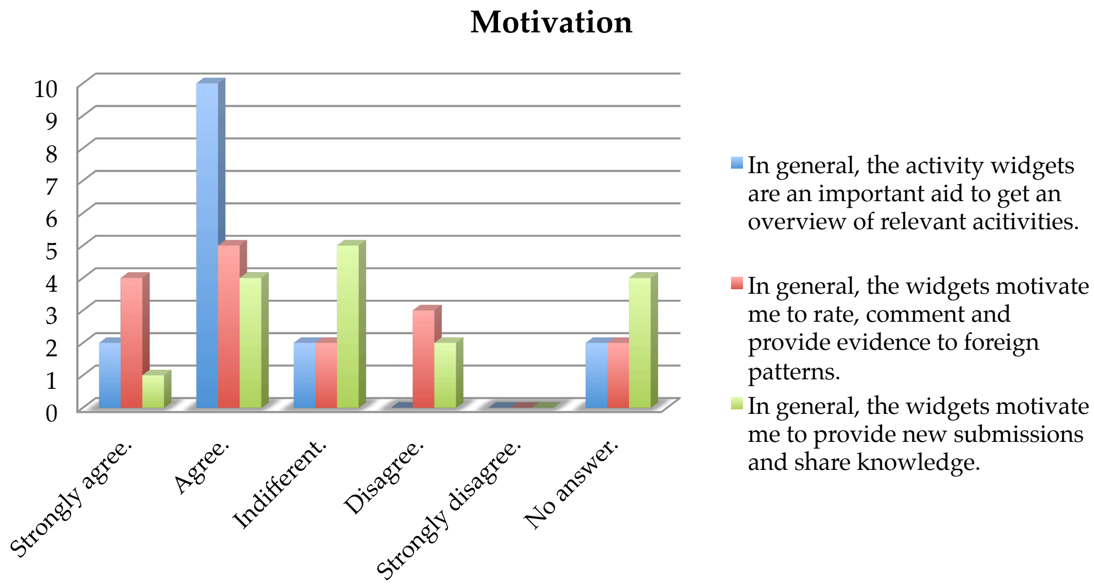


Figure 7.15: Feedback on the effect of widgets on the users' motivation to contribute.

indifferent opinions and disagreements, additional ways to visualize liveliness should be investigated in future work.

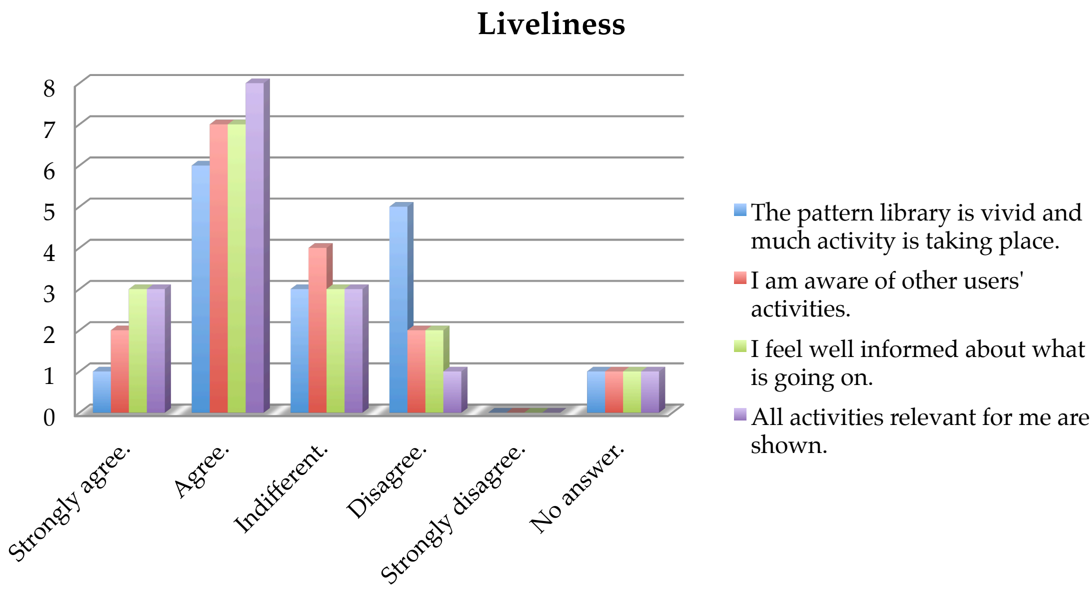


Figure 7.16: Answers on the liveliness of the pattern library.

7.3.4 Usage and Learning Aspects

Additional results to questions dealing with the aspect of time needed to understand the approach are shown in Figure 7.17. The answers indicate that the level of complexity of the implemented prototype was chosen well enough to allow for quickly understanding the EPL concept. However, the need for more training was declared and should be considered when introducing the concept to the project consortium.

Feature complexity

After considering detailed aspects of the presented pattern library, a set of questions asks for the general appeal of the visualization of the library structure as graph and the arrangement of the different widgets. Opinions were collected about the visualization of pattern associations, pattern details and own contributions. Figure 7.18 shows the summarized answers to the seven aspects treated. A general tendency towards all covered aspects can be seen. Stronger indifferences about the graph visualization and pattern details view suggest further improvement of the presentation of these features. First, the overview of user submission may need improvement followed by the graph and association design as well as the widget arrangement. In summary, the overall visualization concept and style was accepted by the participants.

Visualization

Feedback on the general impression of the pattern library concept, its realization and potential for embedment into working processes was inquired. Figure 7.19 reveals that

Users found starting points

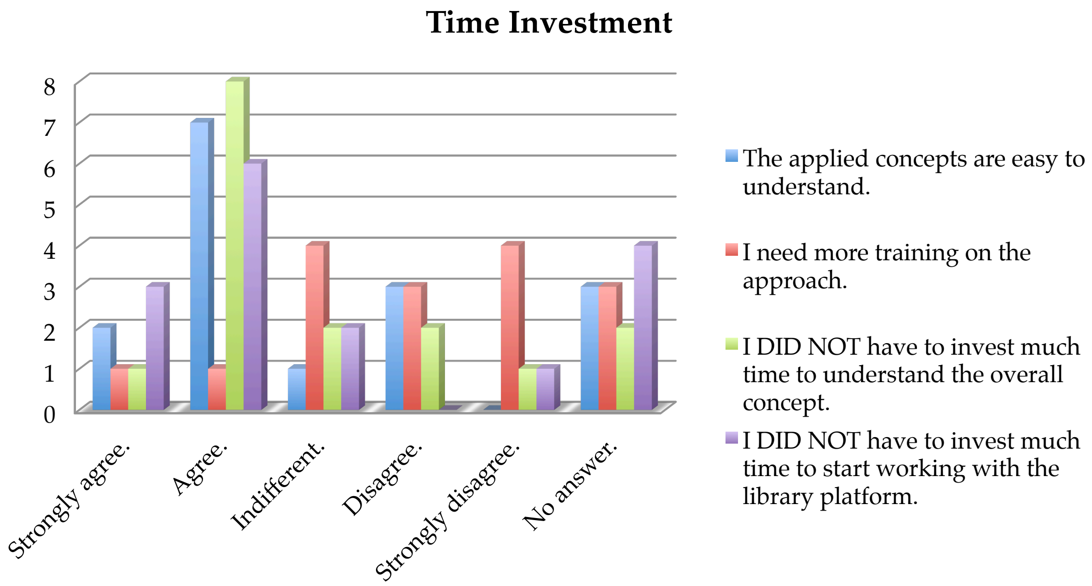


Figure 7.17: In general, the implemented EPL concepts was considered as easy to understand. Little time was needed for familiarization.

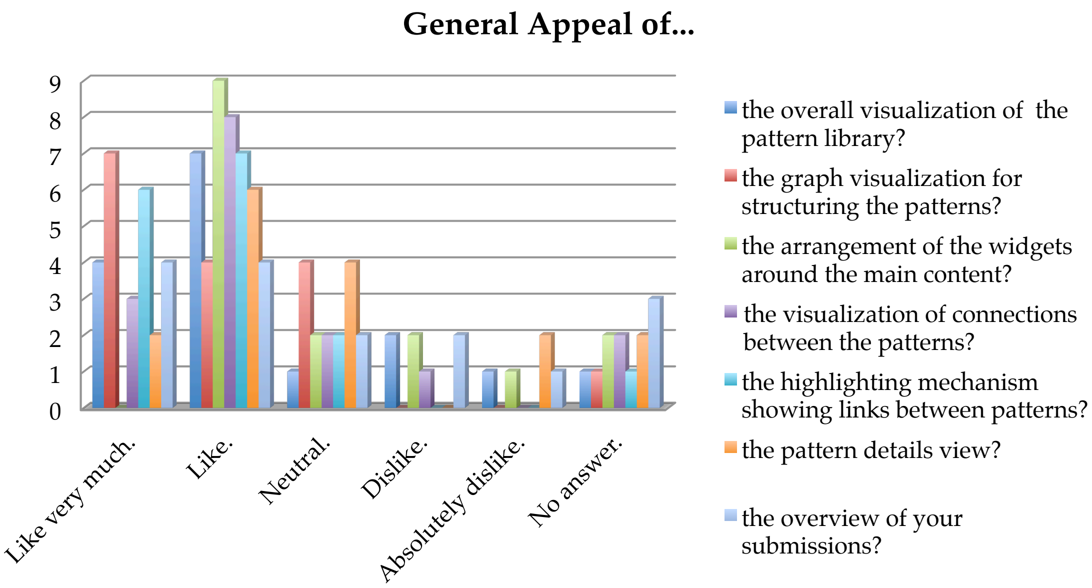


Figure 7.18: The appeal of the presentation of the pattern library and widgets.

most participants could find a starting point for familiarizing themselves with the platform and given information. 2 indifferent opinions and 2 disagreements as well as 2 not given answers indicate that this aspect should be improved in future work. More reading guidance was strongly demanded by 2 participants. 5 answers indicated uncertainty about this aspects, 2 participants did not answer and 6 did not consider more help as necessary. Among them, 1 strongly disagreed with further assistance.

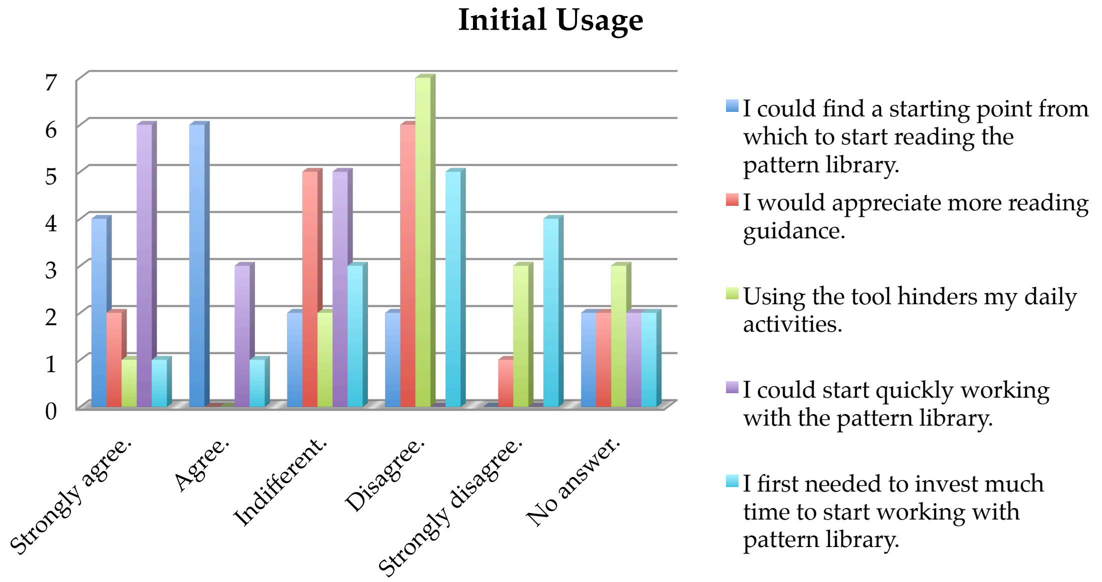


Figure 7.19: Starting points and demand for more assistance.

The presented tool allows every member to contribute to the library at any time. It seems that this thought was affected by the strong guidance of the workshop agenda such that it was hard to estimate the possibility of "freely" using the platform. However, the pattern library was not considered as hindering daily project work by 10 participants. 1 member of the study group stated so, 2 were indifferent and 3 abstained. Refinements of the tool should support its integration into regular project activities. There is potential to use the platform besides the daily business work and on demand. No hindering of daily work

The majority of was able to quickly start working with the pattern library and that most of its features were easy to access and use. However, the amount of abstentions and converse answers suggest to further improve the presentation of content and optimize the interaction concept. Some participants stated that the screen was too "occluded" in some situations and more time was needed to find orientation. Fast start

In general, the evolving pattern library concept was considered as easy to understand by the majority of participants. Figure 7.20 illustrates that 5 members of the study group strongly agreed and 4 ones agreed to this statement. 4 were still indifferent, 1 disagreed, 1 strongly disagreed and 1 answer was not given. Regarding the attractiveness of the Easy to use and browse library, a similar picture is given. Thus, the general direction of illustration seems to satisfy most users. The same statements can be made for the last two questions of that group dealing with the easiness of use of the pattern library and the means to find needed information. Qualitative feedback suggested to improve searching and filtering mechanisms, especially for a growing number of patterns.

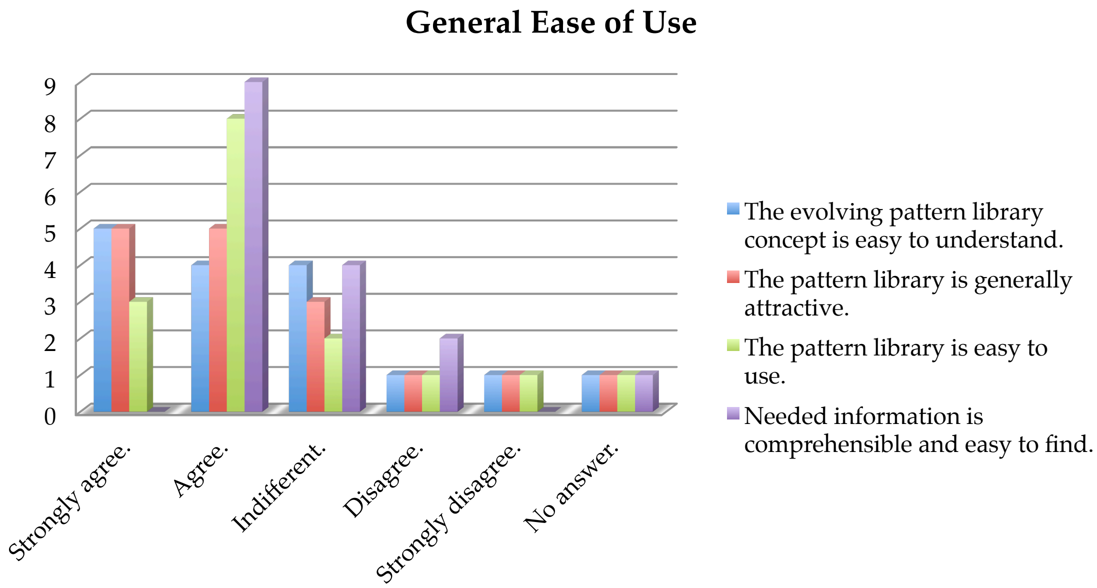


Figure 7.20: Feedback on the ease of use of the pattern library and its attractiveness.

Potential for usage

Questions regarding the general worth of using the pattern library were not definitely answered as illustrated by Figure 7.21. Whereas 8 participants agreed on this statement, 5 opinions remained indifferent, 1 strongly disagreed, and 2 answers were not given. A similar distribution is given for the questions about whether it makes sense to invest time in reviewing and commenting the patterns as well as the future extensibility of the approach. Since only few disagreements were given, it can be concluded that the platform should to be used for longer time periods and as continuous step within the project workflow to obtain clearer results. Project participants may first need to gain more confidence to the approach.

Learning from repository

Figure 7.22 shows the answers given about the participants' estimation on learning aspects. Results look promising since the majority of the study group stated to be able to get better insights into the project in general, and the domain of emergency response in particular. Documentation support that is necessary for filling the pattern library was not directly experience. In its current state, the library was considered positively as knowledge repository from a consuming point of view. The contribution aspect should to be fostered in future iterations. Taking into account qualitative feedback and suggestions given on the aspects of visualization and interaction, primarily means of interaction need to be improved.

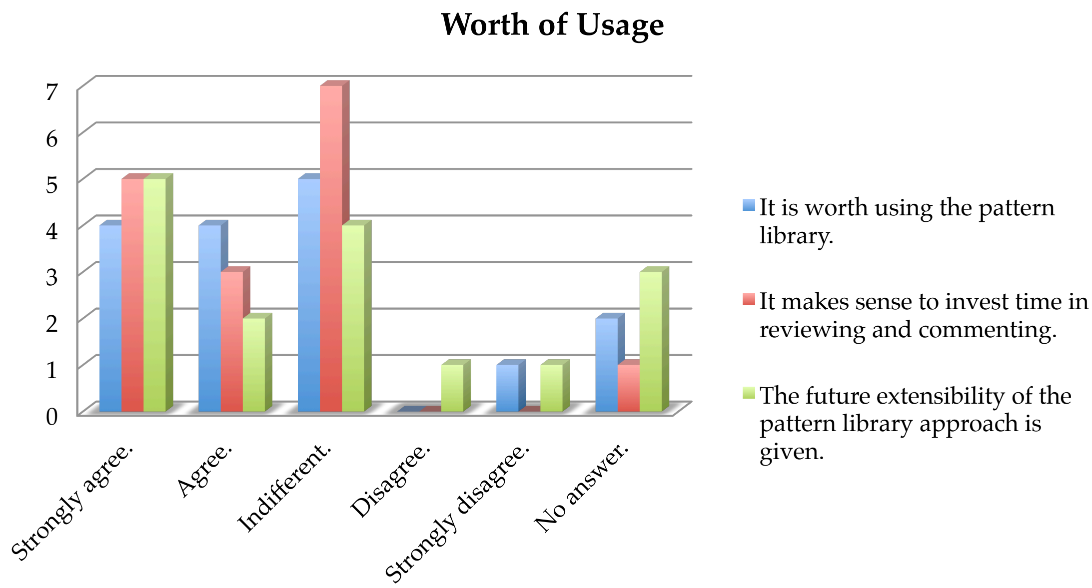


Figure 7.21: Results showing the opinions on the worth of using the platform.

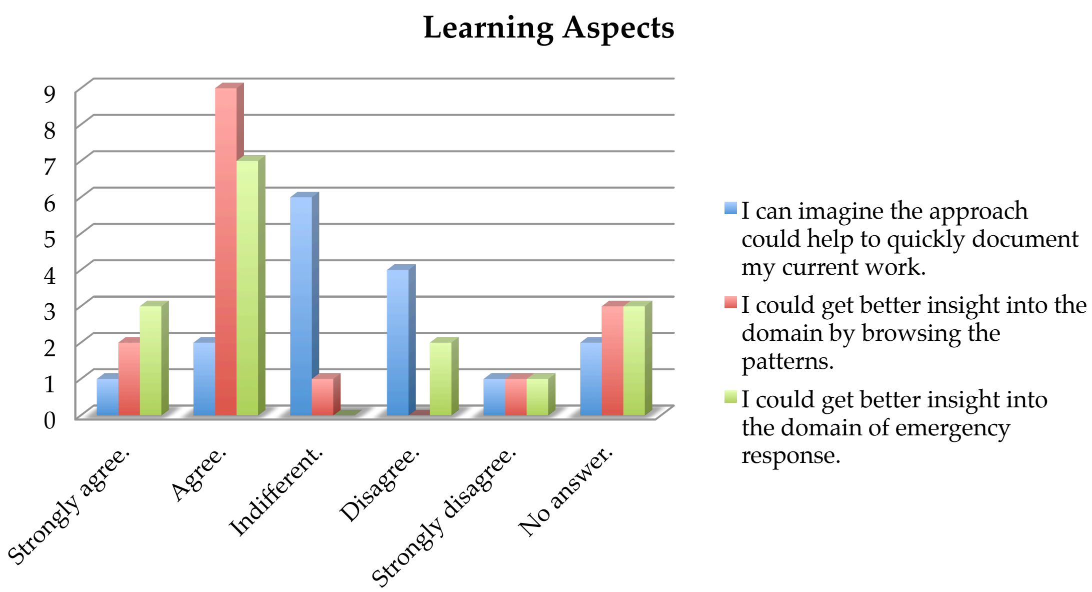


Figure 7.22: Learning from the pattern library contents.

7.3.5 General Feedback and Suggestions

Less technical descriptions

Qualitative feedback from the survey allowed open showed that some participants were concerned on the preparation of information about the EPL concept. Currently, explanations are given in the more technical style as found in this thesis. Suggestions were made to simplify descriptions for rules and roles. State diagrams need a different shape such that other stakeholder with a less technical background gain a better chance of digesting the content. Otherwise, they may be scared off and directly refuse working with the platform. Future iterations should avoid complex descriptions.

Different degrees of freedom regarding the process

Some participants from the second workshop suggested to design the whole structure in a more "open way", like a wiki. Processes should be broken up and more freedom for contributing patterns as well as the modification of existing content should be allowed for every user. For these participants, the role and process model were regarded as too strict and hindering creativity. They described themselves as pattern experts and their work as very creative and often apart from technical development. Interestingly, these concerns were not stated during the first EPL conception phase and the ARL Summer School workshop. A first interpretation of this feedback suggests that the platform's degree of freedom should be adaptable. From the initial workshops and discussions about formulating patterns, potential authors felt overburdened and uncertain on how to begin a pattern formulation. Therefore, the EPL concept introduces the process steps especially to support first orientation and contributions.

Replacement of or addition to established tools

The final part of the questionnaire asked the participants to imagine how to embed the EPL concept into the project workflow and whether it could replace other activities. Answers revealed that the pattern library may be used instead of wiki pages in terms of documenting domain, design and technical knowledge. The envisioned concept of collaboratively exchanging and refining knowledge was considered as promising for future applications. Another statement saw a change for the EPL approach to further support structured knowledge management that was missing in past projects. One participant perceived the presented concept as "boundary object" that adds to the project workflow tool suite instead of a replacing existing ones.

Connection to requirements tracker

Linking the pattern library to a requirements tracking system was another suggestion for future improvement. The documentation field of a requirement could point to appropriate patterns within the library. Contrariwise, patterns from the library could point towards requirements if they describe technical or conceptual restrictions.

Triggers are needed

The course of the studies showed that participants need triggers to start working on documentation. Intrinsic motivation is hard to establish if the immediate working surrounding is not affected. Regarding the need for time management of daily work, this is not a surprise. It seems that the most potential for embedding the EPL approach into a project landscape is at the very beginning of the work. Replacing another step in the workflow may support its acceptance.

7.4 Conclusion

Practical workshop conducted in a research project

This chapter described the practical integration of the EPL approach via the refined prototypes. Rules for quality and validation gateways of the pattern maturation process were configured to illustrate the development of the contributed patterns and the pattern library structure. The joint research project BRIDGE with 60 active participants served as testbed. In total, a representative set consisting of 25% of the project members took part in a distributed pattern collection workshop that spanned several weeks. For the workshop, different assignments aimed at triggering different actions such as reviewing patterns, contributing new one and assigning evidence. However, there were no strict deadlines demanded such that the assignments served as orientation points for task support. The study setup helped to stick to the asynchronous, independent and spatially distributed working habits in a joint research project structure.

Embed process for pattern mining and knowledge management

After the usage period, the pattern library contained 43 patterns in different maturity states. User feedback provided as ratings, comments and the assignment of evidence shows that the EPL concept was implemented in a way that could be used to contribute to the knowledge repository and improve the formulation quality as well as the validity of the different patterns or, as said initially in this thesis, micro-documentations. The fact that most patterns are currently in the states *under consideration* and *pattern candidate* reflects the ongoing project and engineering work. This thesis presents a process for collaboratively gathering and formulating project and domain knowledge within small and quickly readable knowledge containers. In addition, the approach presents an alternative way of mining patterns. Instead of formulating pattern collections in retrospective, proto-patterns and ideas are actively refined by many participants *during* the time of knowledge generation. Looking at the average length of the pattern formulations and the usage of the summary fields, the results are promising.

Library contents as foundation for future work

The set of formulated patterns in the field of emergency response is regarded as basis for future work for documenting project and domain knowledge. The evolving pattern library process lead to promising results and efforts are made to further refine the existing patterns and the structure as well as to include new findings. In a future project, the customized infrastructure and dedicated users in the roles of *Domain Experts* and *Librarians* should be introduced right from the beginning as a part of the development process. The activity of formulating patterns and contributing to their quality and validity should be promoted as early as possible within the project scope.

The study ends but the project continuous

The experimental pursue of the approach ends with the current state of the BRIDGE Pattern Library. Still, the project is running and the library will further accompany it. The results of the survey and the current state of patterns provide confidence in further usage of the platform within the project or adaptations to other settings. User behavior and working phases on the pattern library so far suggest to embed the EPL platform as a project-wide tool and make sure that participants perceive that their contributions are actively maintained, reviewed and consumed.

General understanding of the EPL concept and realization

With the practical usage in mind, all participants answered a substantial questionnaire at the end of the study. The workshop and survey results show that there is potential for the embedment of the evolving pattern library approach as a mean to document knowledge. General understanding and acceptance of the concepts could be revealed. The platform and the concept were considered as easy to access and quickly to use. Although the agenda created a time-lapsed presentation of a real project setting, the thematic connection to the project work of the study group was established.

Thesis summary and outlook

The following chapter gives a summary of the conducted research and engineering efforts together with a summary of the contributions of this thesis. An outlook on future work together with personal remarks conclude this thesis.

Chapter 8

Conclusion and Outlook

This dissertation addresses the challenge of knowledge management in the area of distributed research projects in which a large number of participants with very different backgrounds work together on a future vision. In combination with different specializations, a variety of methods, processes and documentation formats is brought to the overall project scope. Incompatibilities and misunderstandings concerning the used vocabulary and documentation formats are possible side effects of this situation.

Variety of methods, processes and formats

The innovative character of research projects places another burden for the creation, refinement and exchange of knowledge that, in the scope of the regarded projects, is generated by applying iterative and user-centered design methods. Central documentation repositories have to be maintained and common formats must be agreed upon. All work packages within a project need to access and contribute to the common project knowledge. Their achievements, developments and results equally benefit from and contribute to all work packages. A project can only be successful if communication and knowledge exchange accord effectively.

Handling knowledge

The approach presented by this thesis makes use of the concept of patterns by interpreting them as knowledge containers that specifically concentrate on fine-grained pieces of information. Natural language is used for formulating the patterns accepting a potential loss in exactness and precision in favor of wide understandability of the formulated contents. Relations between patterns and the introduction of hierarchies allow the structuring of knowledge from abstract concepts to concrete conceptions and realizations. These *micro-documentations* are collaboratively created, refined and managed by all project members. A process model defines maturity states that reflect the formulation quality and validity of each single pattern. Regarding the exploration of new application domains, it can be important to adapt existing concepts to benefit from preceding work.

Patterns as micro-documentation

8.1 Contribution Summary

Recalling the performed activities

Starting with the motivation of improving the project-related knowledge management processes, the occurring problems regarding knowledge exchange and documentation were analyzed. The established concept of design patterns was further explored and currently existing pattern mining processes were examined. In a user-centered design process, problems for the application of existing concepts were extracted in expert workshops from the domains of emergency response and service-oriented architectures. Based on the identified problems, two sets of requirements were formulated and fed into the conception of the evolving pattern library process. An initial prototype covered a basic set of requirements and led to the extraction of a first set of patterns. A second iteration implemented the advanced set of requirements. The refined prototype was accepted by the participants of pattern collection and acceptance study workshops throughout which the realized tool was actively used. The following sections describe the three main contributions of this thesis that cover the conception of an evolving pattern library, its technical realization as a feasibility study and the derived pattern library in the scope of a concrete research project.

8.1.1 Evolving Pattern Library Concept

Iterative refinement of formulations

The first contribution provides a generalized concept of a collaborative formulation process for evolving patterns within a dynamic library . The process covers documenting, sharing, refining and reusing project-related knowledge for domain analysis, application, interaction and system design as well as sociological, ethical and legal issues. Validation, demonstration and dissemination tasks benefit and contribute their results and feedback to the artifacts created throughout the process. Over the project's lifetime, new experiences, findings, ideas and evaluations are continuously fed back into the pattern library. All project-related ideas and findings are formulated as early proto-patterns. Time, discussion and validation show whether the idea is a successful pattern, a surprisingly failing approach leading to an anti-pattern or if the early formulation turns out as not suited to the project-related knowledge.

Early involvement

The collaborative pattern formulation process involves a *large group* of potential pattern authors and reflects the iterative design approaches that are often characteristic for ICT-related research projects. The concept considers every project member's experience and ideas as a potential contribution to the pattern library. Since the presented approach allows knowledge sharing and development during the project as soon as possible, the threshold for participation is kept low in order to make the pattern library vivid and dynamic. The involvement of the whole community intends to support the liveliness of the pattern language and the amount of collected patterns. This changes the derivation approach of a pattern language from the expert's point-of-view to a mixed approached in which experts and novices can benefit from each other. Expert knowledge can already be formulated as approved patterns. Additionally, ideas and concepts can be brought in as pattern candidates. The community-based validation process then carves out whether a candidate actually describes a pattern or an anti-pattern.

Role model for contribution and management

Contributed patterns, feedback and suggestions are managed via a light-weight role model that allows to differentiate between *Visitors* of the pattern library who only have reading access to the knowledge repository, *Members* and *Authors* that contribute feedback and formulate patterns, and *Domain Experts* that assure the submission quality and support authors in validating provided evidence and incorporating suggestions for reformulations. *Domain Experts* are organized in the *Submission Control Board* such that the work can be shared and decisions are made in a democratic way. The special role of a *Librarian* takes a more administrative responsibility for the management of published patterns and their sustainability as well as structural changes and supervision of the pattern library. Multiple librarians adjust their activities within a special *Structural Advisory Board* for sharing the workload and taking group-based decisions. The sizes of the advisory boards are customizable for the current project's scope. All parameters for determining a pattern's formulation quality and validity as well as the composition of advisory boards remain flexible and can be changed during the project lifetime.

Pattern maturation process

A maturity model for the individual pattern formulation is specified that first focuses on the formulation quality of the pattern content itself. Over time, the pattern is validated by collecting evidence factors that support or refute the proposed solution. The maturity model allows for early formulations of open problems for which solutions need to be found. This state reflects the ongoing research work starting from an identified problem.As soon as a pattern's name, problem and solution can be formulated, the maturity process accompanies the proto-pattern *under consideration* during its maturation to a *pattern candidate* that is finally validated to an *approved pattern*.

Integration of anti-patterns

Surprisingly failing design concepts are documented as anti-patterns as described in preceding work (cf. Reiners et al. [2011]). Anti-patterns represent knowledge about design and conception flaws that should not be repeated. Resources regarding research and development can be optimized by concentrating on promising concepts. The design space can be explored further in new directions bypassing existing knowledge about successes and failures. The presented approach allows the development of a formulated pattern into an anti-pattern and vice versa, based on collected evidence.

Flexible hierarchy of pattern categories

Hierarchy levels categorize and structure patterns starting from the description of abstract rules and processes up to more concrete concepts and designs. New hierarchy levels can be introduced on demand into the structure. The pattern library is organized as a directed, acyclic graph according to the pattern language definition given by Borchers [2001]. The iterative and dynamic character of distributed research projects is reflected by the continuous formulation of new and the assessment of existing patterns that serve as micro-documentations and the flexible structure of the pattern library itself. New patterns are continuously integrated and new associations between the patterns are established. Accordingly, it may become necessary to change existing associations and to refactor existing patterns.

8.1.2 Technical Infrastructure

Customizable open source pattern repository

The second contribution is a generic technical realization based on an open source content management system. All parameters concerning the quality gates for the pattern formulation quality and validity are customizable according to a concrete project scope. The role model was implemented alongside with the maturity states and hierarchy levels proposed by the concept. The technical platform serves as practical testbed for the application of the theoretical concept of an evolving design pattern library formulation process. Two case studies in the domain of emergency response and service-oriented architectures showed that the general concept was understood by participants and the flexibility of the quality and the rule system found acceptance.

Structural visualization as DAG

Concepts for showing the pattern library structure follow the discussions from early expert workshops during which different visualization methods were discussed. As overall result, an overview of the whole pattern library structure in a graph-like shape was demanded from which detailed views of the single patterns are accessible. Navigation between the patterns is realized via links.

Metaphors for pattern maturity

Concerning the maturity states of the individual patterns, different metaphors were proposed and discussed. Common to all metaphors was the quality of showing development of the knowledge item. Thus, a plant's growth states, different moon phases, development states of a butterfly and developing cities were proposed.

Personalized information

The dynamics of the structure concerning new, changed and orphaned patterns together with associations were incorporated into the technical infrastructure. A personalized dashboard shows the current state of user submissions in the role of an author. Comments and evidence provided by members together with the overall activity of the platform are presented as summaries. For *Domain Experts* and *Librarians*, specialized views on open tasks concerning the quality check of new contributions and acceptance of proposed evidence are implemented.

Track changes

Mechanisms to track changes to reproduce reasons for decision and monitor user activities are part of the provided system. This way, callbacks for submissions can be realized. Over time, *Members* and *Authors* may be promoted to *Domain Experts* asked to join the *Submission Advisory Board*. Outdated patterns can equally be determined.

Integration into workflow

Potential for embedding the presented approach within the life cycle of research projects as a repeated process step was indicated. The technical realization then adds to existing productive tools with the specialized purpose of knowledge generation and exchange.

8.1.3 Derived Pattern Library

Revealed patterns from case study

In order to validate the derived concept, the technical platform was continuously used in the scope of the project BRIDGE that served as main test bed for validating the approach. According to its goal, new technologies and approaches for large scale emergency response and crisis management are derived, developed and validated. The abstract design knowledge and general guidelines for system design in this domain could captured and updated during the pattern collection workshops and represent a beneficial outcome of the engineering work performed for this thesis. The BRIDGE Pattern Library currently[1] consists of 43 patterns that underwent the quality assessment and validation process in different stages.

Pattern maturity distribution

2 *approved patterns* could be mined during the project work so far. 17 patterns are considered as *pattern candidates* and 24 patterns are currently *under consideration* meaning that the formulation quality still needs to be improved before an evidence-based validation check is conducted. Research and development activities can be described by the formulation of open problems. As soon as new concepts and results are available, they will be added by the responsible author or as comments from project members that work in the problem scope.

User needs as patterns

During the pattern collection workshops , it turned out that especially high-level patterns, i.e., general rules and processes, were implicitly pre-formulated as non-technical user demands. They were regarded as essential to be considered when designing a system but are mostly ignored by work packages concerned with the technical implementation. There was a perceived misplacing of these demands that were discusses during the domain analysis phase. With the introduction of the pattern library concept and the transformation of user demands into high-level pattern formulations, this conflicting situation could partially be solved.

Pattern category depends on work package

In summary, many patterns were placed in the higher, more abstract levels of the hierarchy by members of work packages concerned with domain analysis and social, legal and ethical issues as high-level concept rules and guidelines. Participants from the work packages that were more related to interaction and application design, implementation as well as demonstration, validation and dissemination primarily contributed to the middle and lower hierarchies of the pattern library. This result reflects the primary focus of the individual work packages. All patterns that were collected in the scope of the project BRIDGE were assembled into an online available pattern library[2].

8.2 Future Work

New aspects

While engineering the approach, especially during the concept design and pattern collection workshops, several ideas came up. Not all could not be handled within this thesis but represent interesting opportunities for extensions and future work.

Ranking and reputation

The evolution of the individual participants concerning their occupied roles based on their experience and contributions should be explored in future approaches. In combination with mechanisms for building a member's reputation, specializations could be

[1]State: September 2013.
[2]http://pattern-library.sec-bridge.eu

made more prominent. Automatically changing roles depending on an internal scoring mechanism should be taken into account. This way, the individual users' activities may become more visible to others and motivate them to achieve similar reputation.

Authorship

In combination with contributions and reputation, questions concerning the authorship of a pattern were raised. Future approaches should incorporate mechanisms for shared content ownerships. This thought gains importance regarding the promotion of user activities. Links to related work formulated by Prause [2013] are seen in combination with the developed *CollabReview* platform. It considers aspects of code ownership and ensuring quality for software artifacts.

Incentives to increase motivation

Concerns were stated that contributors might loose motivation of performing their tasks concerning the contribution and validation. Similar considerations are presented by Prause [2013] in the scope of documentation quality of software artifacts. As a solution, incentives that address intrinsic and extrinsic motivational factors were presented together with a reputation system based on scores for the participant's contribution and review activities. Previous work explored the incentives space that could be used as a foundation for combining the concepts with the EPL approach (cf. Prause et al. [2010a]). For the Yahoo! pattern formulation process, Leacock et al. [2005] present first ideas to motivate potential authors by a raffling. In his approach, users receive tickets upon every contribution, rating activity and defining quarterly goals. Discussions held on that topic showed that the latter two ideas were considered as very competitive and might frighten off potential authors. Such an effect would be counterproductive to the initial idea of the EPL approach which explicitly encourages early formulations that are refined collaboratively with the support of other users.

Further research on anti-patterns

Elaborations on the concept of anti-patterns could not be further pursued in the scope of the workshops. It seems that the derivation of anti-patterns requires more time for research validation. Long-term studies should analyze this topic and the correlation of supporting and refuting evidence, respectively. Additionally, an appropriate visualization of anti-patterns as addition to an accepted maturity state metaphor needs to be investigated, eventually depending on the current project's scope.

Extraction of collections

Further dissemination of the project results can be achieved by extracting mature patterns as prominent pattern collection. More sophisticated categorizations become possible that extract the most mature and best proven patterns on domain-related knowledge according to rules, laws and established processes. This way, project externals who should be informed about the project's achievements may use the elaborated findings of the project. Alternatively, the best rated and matured interaction concepts or project-related technical outcomes could be assembled. The latter idea might be interesting especially for dissemination and exploitation activities that are another important factor for the whole research project's achievements and outcomes.

Addition of semantics

In the scope of the presented work, the derivation of a process and the provision of a technical infrastructure that can be modified according to the need of different projects' were regarded as important contribution. The library concept creates a foundation for working with patterns as evolving knowledge containers. Future extensions should consider making use of ontologies that automatically identify related patterns and take into account different semantics of pattern associations. Semi-automated proposals of pattern sequences based on queries could then be realized. However, the concept presented so far concentrates on *human collaboration* and the manual exploration of project knowledge to be understood and internalized by the project members. Once the pattern library contents grow and patterns alternatives emerge, an approach as described by Zdun [2007] may support the selection of adequate patterns. The idea is to formalize pattern relationships by applying reusable annotated grammars to extract relevant patterns that match defined quality criteria.

Concept transfer to industry

Especially during the workshops held ARL Summer School event at Reutlingen University, representatives from industrial companies showed interest in the concept of gradually capturing and continuously updating small pieces of information. For them, it was very important to establish a light-weight knowledge repository that is maintained by a core team but remains open to contributions from employees of many departments. Transferring and customizing the presented approach to industrial environments appears as promising idea worth exploring in conjunction with the elicitation of new requirements for the EPL process. Discussions came up about linking patterns to architectural decisions and recommendations.

8.3 Closing Remarks

Potential of patterns

Throughout the development of the presented approach and active discussions on the topic during workshops and conferences, the potential of the pattern concept became visible. The application of patterns in a variety of domains and contexts strongly encourages the integration into research and industrial development settings. They stand at the threshold for being used in a much more ubiquitous and collaborative way beyond sharing and rating as currently performed in online catalogs. Pattern management tools and automatized recommending systems are another interesting field to be explored, especially if patterns are predominantly used as solution approaches to problems in a knowledge repository. The understanding of knowledge goes further and has the intention to educate participants as well as to learn from them in a collaborative manner.

Acceptance depends on willingness and integration

An important result for the successful establishment of the evolving pattern library concept is the willingness of the project consortium to accept this kind of documentation. In the beginning, the library should already be seeded by a core team with existing patterns and ideas originating from prior experience. A community-based approach within a research project, to which every member continuously contributes on a voluntary basis, is hard to realize. In order to improve its usage, the approach should be established right from the start of the project and be embedded into the project workflow such that it becomes a settled project-related activity. In other settings, like free crowd-based projects in which many volunteers collaborate on a common knowledge repository as in wikis or online pattern collections, the process may have better chances of being quickly accepted and integrated into existing practices.

Communication remains a human matter

Knowledge management and communication are important drivers for the success of research institutions and enterprises, but also for the personal development. Not everything needs to be conveyed electronically. The personal exchange during business work and common leisure activities as well in the famous kitchen corner of a department must never be neglected. Apart from all processes and concepts that are invented, research and development are still performed by humans. Therefore, technology can only support human activities and behaviors but never replace them.

Appendix A

The Introductory Flyer for the BRIDGE Pattern Library

The BRIDGE Design Pattern Library

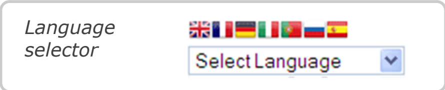
What is a design pattern?

- A good solution to a common design problem within a context
- Preservation of timeless ideas and concepts → Reusability
- Described by text and illustrations → Readable by non-experts

A design pattern can be seen as a "best practice" design element

Example of a simple design pattern

- **Name:** Language selector
- **Context/problem**: On a multinational website, users need a simple way to select their preferred language
- **Description/solution**: use a box where you can click on a flag or select nationality from a pull-down-menu
- **Illustration:** see example graphics



Key elements:
- Name
- Context/problem
- Description/solution
- Illustration

What is the BRIDGE Design Pattern Library (DPL)?

- A web based library where we collect design solutions developed within BRIDGE or imported from other sources.
- Design patterns for crisis management systems will be our target, and key elements like e.g. role, device, and task will be added to the more standard elements of a design pattern.
- The BRIDGE DPL will be an "empty" library until we fill it with content together throughout the project lifetime.
- As a user of the library, you can browse and comment on existing patterns, or submit new patterns to the library.
- New patterns will be discussed and validated through WP 9 (demonstration) and WP 10 (validation).
- Deliverable D06.1 will be the library structure in itself. Please follow our progress here: link to the BRIDGE DPL.

The pattern library will grow as we fill it with content throughout the project lifetime.

Draft version, please visit http://bridge-pattern-library.fit.fraunhofer.de/

What we need from you:

- We are currently building the library structure, and we need your input and suggestions on how it could serve BRIDGE in an optimal way.
- More specifically: we need your detailed suggestions on features and functionality – how the library should work.
- We also need design patterns that you already have, to prevent the library from being empty at the delivery date.
- We need your feedback as soon as possible, as the library will be handed over for internal review by the end of November.

We need feedback and pattern suggestions

- First draft: Oct. 21
- Second draft: Oct. 28
- First review: Nov. 20
- Delivery date: Dec. 31

Where can I get help?

- Please contact the DPL team via the e-mail group *Design Pattern Library* in the eRoom (link to e-mail groups)
- Get more details from the DPL handbook, which will be continuously updated in the eRoom (link to handbook).
- See also our collection of design pattern literature on Mendeley (link to literature)

Please follow our progress and give us your feedback on the way.

Oct 14, 2011 – a flyer from the Design Pattern Library team: Ragnhild, Aslak, René, Daniela

Appendix B

Questionnaire Used for Preparing the Refined EPL Prototype

Thank you for taking part in the survey!

The BRIDGE pattern library represents a prototypical implementation of a collaborative pattern formulation approach that extends the traditional pattern mining and formulation concept. The major goal of is to collect requirements for improving visualization of the library and collect new ideas for further features.

The results are anonymized. Please keep in mind that the current system and future ideas are assessed - not your answers or interaction with the current system.

Participant's Background

1. What is your professional background?

2. How old are you?

3. What is your level of knowledge about design patterns?

None. ( ) Little. ( ) Average. ( ) Good. ( ) Expert. ( )

4. In what context did you, if applicable, use design patterns?

5. Current Features Walkthrough

Please visit the BRIDGE's design pattern library online and browse the available patterns. Follow the url below to visit the website:
http://pattern-library.sec-bridge.eu

6. How do you like the current browsing mechanism?

Strongly dislike. ( ) Dislike. ( ) Indifferent. ( ) Like. ( ) Strongly like. ( )

7. How do you like the presentation of pattern details?

8. The BRIDGE design pattern library is light-weight.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

9. The BRIDGE design pattern library is too light-weight.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

10. The platform is easy to understand.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

11. The patterns are easy to browse.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

12. What features are mainly missing in the current version of the pattern library?

13. Do you want your activities to be seen by other users?

Yes ( ) No ( )

14. An activity stream is useful to show activity within the pattern library portal.

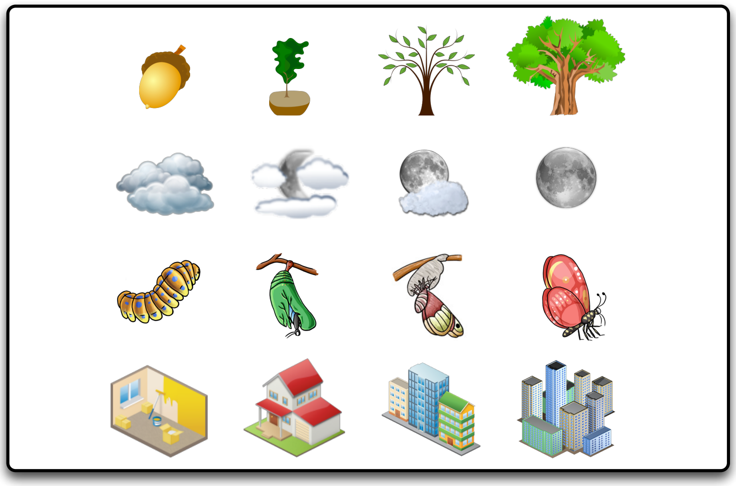
Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

15. What additional information would you like to be shown in the activity stream?

16. How do you rate the current means of showing activities in the pattern library?

Poor. ( ) Fair. ( ) Indifferent. ( ) Good. ( ) Very good. ( )

Discussion of a Metaphor for Visualizing a Pattern's Maturity We have considered using some metaphors to represent the structure of the design pattern library and its dynamics. We have come up with some initial metaphors. You can find the different icons below. The leftmost icons represent the least mature state of a pattern. Reading from left to right, the maturity states represent *open problems*, i.e., pattern ideas without solutions, patterns *under consideration*, *pattern candidates* and *approved patterns*.



17. Please provide a ranking of the proposed metaphors from 1 to 4.

Moon phases. ( )
Stages of a plant's growth. ( )
Evolution states of a butterfly. ( )
Development of a city. ( )

18. Do you have additional ideas for the presented metaphors or do you have further ideas for metaphors that represent a pattern's maturity?

19. A pattern's maturity should be visualized by a metaphor.

Strongly dislike. ( ) Dislike. ( ) Indifferent. ( ) Like. ( ) Strongly like. ( )

20. Do you have any other suggestions or comments about the current approach or future improvements?

Appendix C

Post-Study Questionnaire: The ARL Summer School Workshop

Welcome to the survey on the Evolving Pattern Library (EPL) Tool presented during the ARL Summer School Workshop at Reutlingen University. After you have been introduced to the general EPL concept and used the actual tool, we would like to know from you about your impressions, ideas and opinions on the concept itself and its partial realization via the tool.

It is the tool which is tested and evaluated - and NOT you. So, please feel free to provide feedback on things you liked or where you see changes for improvements.

Thank you very much for taking part!

Participant's Background

1. What is your background?

2. What is your current occupation?

3. How old are you?

4. We would like to know more about your experience with design patterns.

In what context did you use design patterns?
Please let us know if you mainly used them as reference or if you have also written patterns or contributed to pattern writing.

5. What is the level of knowledge you possess about design patterns?

Expert. ( ) Good. ( ) Partial. ( ) Rough. ( ) None. ( )

Rodel Model

6. The purpose of the different roles is understandable.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

7. The concept of advisory boards is understandable.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

Pattern Maturity Concept

8. The concept of the pattern maturity is understandable.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

9. The rules for maturity state changes are understandable.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

10. The used metaphor is easy to understand.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

11. Do you think that states were missing? Which one(s)

12. Do you think that there were obsolete maturity states? Which one(s)?

13. Do you have any suggestions regarding the metaphor?

Hierarchy Concept

14. The concept of pattern hierarchies is understandable.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

15. Do you think that hierarchy levels were missing? Which one(s)?

16. Do you think too many hierarchy levels were introduced? Which one(s)?

Understandability and Usage

17. The evolving pattern library concept is easy to understand.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

18. The pattern library is generally attractive.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

19. The pattern library is easy to use.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

20. The pattern library is easy to browse.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

21. Needed information is comprehensible and easy to find.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

22. It is worth using the pattern library.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

23. It makes sense to invest time in reviewing and commenting.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

Contributing and Rating

24. Submitting new patterns was easy to perform.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

25. The rating on the pattern formulation is easy to perform.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

26. Adding new evidence and comments is easy.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

27. The evidence classification is easy to understand.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

28. The current platform provides a set of evidence type like, e.g., "existing products" or "Research Publication". Are there other kinds of evidence you would like to include? Which one(s)?

29. Which kinds of evidence do you regard as obsolete, if any?

Visualization

You have tried out the EPL prototype by browsing the library, rating existing patterns and contributing your own patterns. Additionally, new evidence was provided by you.

30. How do you like the overall visualization of the pattern library?

Very much. ( ) Good. ( ) Indifferent. ( ) Dislike. ( ) Strongly dislike. ( )

31. How do you like the graph visualization for structuring the patterns?

Very much. ( ) Good. ( ) Indifferent. ( ) Dislike. ( ) Strongly dislike. ( )

32. How do you like the presentation of pattern details?

Very much. ( ) Good. ( ) Indifferent. ( ) Dislike. ( ) Strongly dislike. ( )

33. What features are missing in the current visualization? What should be improved?

Activity and Liveliness

34. The pattern library is vivid and much activity is taking place.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

35. The provided widgets ongoing activities in an understandable way.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

36. Are there missing widgets? What kind of?

37. Are there obsolete widgets? Which one(s)?

38. The future extensibility of the pattern library approach is given.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

39. How would you judge the usefulness potential of approach?

Very high. ( ) High. ( ) Moderate. ( ) Low. ( ) Very low. ( )

40. There is high potential for the integration of the evolving pattern library concept into a project workflow to improve knowledge management.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

41. Do you think the approach could replace processes currently applied in projects? If yes, which one(s)?

42. What features are missing in the current concept? What should be improved?

43. Do you have any additional comments?

Appendix D

Post-Study Questionnaire: The BRIDGE Pattern Workshop

Welcome to the survey on the technical realization of the Evolving Pattern Library (EPL) concept that was developed in the scope of the BRIDGE project!

After you have been introduced to the general EPL concept and used the actual tool, we would like to know from you about your impressions, ideas and opinions on the concept itself and the partial realization via the tool.

While answering the questions, please keep in mind that it is the tool and concept which is evaluated NOT you. You are the one helping us to find the flaws and missing features in the tool and concept. Therefore, feel free to provide feedback on things you liked or where you see chances for improvements.

Thank you very much for taking part!

Participant's Background

1. What is your professional background?

2. What is your primary activity in your current main project (e.g., domain analysis, interaction design, architecture etc.).?

3. What other activities do you perform in the scope of your main project?

4. Is your position primarily related to research or industry (or both)?

5. How old are you?

Pattern Experience

6. What is the level of knowledge you possess about design patterns?

Expert. ( ) Good. ( ) Partial. ( ) Rough. ( ) None. ( )

7. How would you judge your expertise in the domain of emergency response?

Expert. ( ) Good. ( ) Partial. ( ) Rough. ( ) None. ( )

8. What is your level of expertise in software engineering?

Expert. ( ) Good. ( ) Partial. ( ) Rough. ( ) None. ( )

9. What is your level of expertise in user-centered design?

Expert. ( ) Good. ( ) Partial. ( ) Rough. ( ) None. ( )

10. How often do you make use of design patterns?

Not at all. ( )
Rarely (once a year). ( )
Sometimes (at least once per half year). ( )
Often (at least once per two months). ( )
Regularly (at least once per month). ( )

11. How often have you written custom a design patterns BEFORE using the EPL tool?

Not at all. ( )
Rarely (once a year). ( )
Sometimes (at least once per half year). ( )
Often (at least once per two months). ( )
Regularly (at least once per month). ( )

12. How often do you get in touch with design patterns in your daily work?

Not at all. ( )
Rarely (once a year). ( )
Sometimes (at least once per half year). ( )
Often (at least once per two months). ( )
Regularly (at least once per month). ( )

13. In what context did you use design patterns?

Role Model

14. The purpose of the different roles is understandable.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

15. The concept of advisory boards is understandable.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

16. It is clear to me when my submissions are reviewed by the advisory board.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

17. I understand why pattern and evidence submissions are not directly available (only after internal review)

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

18. Do you have any additional comments regarding roles?

Rating

19. The rating on the pattern formulation is easy to perform.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

20. The meaning of the different aspects is clear.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

21. It is easy to understand that a minimal number of ratings needs to be performed to trigger the check for a pattern's maturity advancement.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

22. The average rating provided by others is clear to me.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

23. The consequences of my rating for an individual pattern are clear.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

24. It is easy to understand that rating is performed to improve the pattern's formulation quality.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

25. It is clear that I can change my rating and only my last rating is taken into account for the formulation quality assessment.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

26. It is easy to add comments.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

27. It is clear that comments are tied to pattern maturity states to keep track of the pattern's formulation development.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

28. Do you have any comments regarding the rating and commenting functionality? What is missing? What can be improved?

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

Evidence Assignment

29. Adding new evidence is easy to perform.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

30. The evidence classification is easy to understand.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

31. It is clear that I can choose from existing evidence and add new entries.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

32. I understand that assigned evidence is checked by the submission control board first.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

33. The concept of supporting and refuting evidence for a pattern is clear to me.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

34. It is clear that collected evidence affects a pattern's reliability towards an approved pattern or towards and anti-pattern.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

35. Are there other kinds of evidence you would like to include? Which one(s)?

36. Which kinds of evidence do you regard as obsolete, if any?

37. Do you have any comments regarding the notion of evidence provision? What it missing? What can be improved?

Usage

38. The design pattern library is easy to browse.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

39. The pattern library platform is easy to understand.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

40. Submitting new patterns is easy to perform.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

41. Changing my own submissions is easy to perform.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

Pattern Maturity Concept

42. The concept of the pattern maturity is understandable.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

43. The rules for maturity state changes are understandable.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

44. The meaning of the different maturity states is clear to me.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

45. Do you think that states are missing? Which one(s) and what should they express?

46. Do you think that there are obsolete maturity states? Which one(s)?

Maturity Metaphor

47. The pattern maturity metaphor is easy to understand.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

48. The used metaphor icons are appropriate.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

49. I like the used maturity icons.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

50. Do you have any suggestions regarding the metaphor? What is missing? What can be improved?

Pattern Hierarchy Concept

51. The concept of pattern hierarchies is easy to understand.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

52. The meaning of the different hierarchy levels is clear to me.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

53. I had problems assigning patterns to an appropriate hierarchy level.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

54. I understand why existing patterns are assigned to a specific hierarchy level.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

55. It is clear that the upper hierarchy levels represented more abstract concepts than the lower ones.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

56. The graph helps me to spot where new knowledge could be provided.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

57. Do you think that hierarchy levels are missing? Which one(s)?

58. Do you think there are too many hierarchy levels? Which one(s) are unnecessary?

Tracing Submissions and Activities

59. It is easy to follow activities related to my patterns.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

60. It is easy to follow activities related to my evidence submissions.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

61. I always know the state of my submitted patterns.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

62. I feel well-informed about the review and publication status of my submissions.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

63. What features are missing in the current visualization? What should be improved?

Widgets

64. The provided activity stream shows ongoing activities in an understandable way.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )
Not noticed. ( )

65. The newsticker helps me to gain an overview of activities related to my submissions.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )
Not noticed. ( )

66. The widgets showing needed ratings and evidence motivate me to collaborate on these patterns.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( ) Not noticed. ( )

67. The hall of fame motivates me to take part in the collaborative pattern library process.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( ) Not noticed. ( )

68. The hall of fame provides a good overview that other users are collaborating as well.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( ) Not noticed. ( )

69. The overview of still incomplete patterns is a good way to suggest me what to do next.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( ) Not noticed. ( )

70. The widget about users who are online shows me that I am not solely working on the platform.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( ) Not noticed. ( )

71. Are there missing widgets? Which one(s) would you like to be added?

72. Are there obsolete widgets? Which one(s)?

73. Do you have any general comments or suggestions regarding the presentation of activity and liveliness? What is missing? What could be improved?

Motivation and Liveliness

74. The pattern library is vivid and much activity is taking place.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

75. I am aware of other users′ activities.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

76. I feel well informed about what is going on.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

77. All activities relevant for me are shown.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

78. In general, the widgets motivate me to rate, comment and provide evidence to foreign patterns.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

79. In general, the widgets motivate me to provide new submissions and share knowledge.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

Time Investment

80. The applied concepts are easy to understand.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

81. I need more training on the approach.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

82. I DID NOT have to invest much time to understand the overall concept.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

83. I DID NOT have to invest much time to start working with the library platform.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

Visualization and Interaction

84. How do you like the overall visualization of the pattern library?

Very much. ( ) Good. ( ) Indifferent. ( ) Dislike. ( ) Strongly dislike. ( )

85. How do you like the graph visualization for structuring the patterns?

Very much. ( ) Good. ( ) Indifferent. ( ) Dislike. ( ) Strongly dislike. ( )

86. How do you like the arrangement of the widgets around the main content?

Very much. ( ) Good. ( ) Indifferent. ( ) Dislike. ( ) Strongly dislike. ( )

87. How do you like the visualization of connections between the patterns?

Very much. ( ) Good. ( ) Indifferent. ( ) Dislike. ( ) Strongly dislike. ( )

88. How do you like the highlighting mechanism showing links between patterns?

Very much. ( ) Good. ( ) Indifferent. ( ) Dislike. ( ) Strongly dislike. ( )

89. How do you like the pattern details view?

Very much. ( ) Good. ( ) Indifferent. ( ) Dislike. ( ) Strongly dislike. ( )

90. How do you like the overview of your submissions ("My Contributions")

Very much. ( ) Good. ( ) Indifferent. ( ) Dislike. ( ) Strongly dislike. ( )

Initial Usage

91. I could find a starting point from which to start reading the pattern library.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

92. I would appreciate more reading guidance.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

93. Using the tool hinders my daily activities.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

94. I could start quickly working with the pattern library.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

95. I first needed to invest much time to start working with the pattern library.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

General Ease of Use

96. The evolving pattern library concept is easy to understand.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

97. The pattern library is generally attractive.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

98. The pattern library is easy to use.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

99. Needed information is comprehensible and easy to find.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

Worth of Usage

100. It is worth using the pattern library.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

101. It makes sense to invest time in reviewing and commenting.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

102. The future extensibility of the pattern library approach is given.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

103. Do you think the approach could replace processes currently applied in projects? If yes, which one(s)?

Learning Aspects

104. I can imagine the approach could help to quickly document my current work.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

105. I could get better insight into the domain by browsing the patterns.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

106. I could get better insight into the domain of emergency response.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

107. There is high potential for the integration of the evolving pattern library concept into a project workflow to improve knowledge management.

Strongly agree. ( ) Agree. ( ) Indifferent. ( ) Disagree. ( ) Strongly disagree. ( )

108. What features are missing in the current concept? What should be improved?

109. Do you have any additional comments on the EPL approach, the provided platform, visualization or interaction?

Appendix E

An Evolving Pattern Library for Emergency Response

Currently, the BRIDGE Pattern Library contains 43 patterns and is structured via the hierarchy levels as proposed in Section 6.2.1. 24 patterns are in the state *under consideration*, 17 are considered as *pattern candidates* and 2 reached the state of an *approved pattern*. The following sections present the current state of the publicly available BRIDGE Pattern Library[1] at submission time and are structured according to the pattern library hierarchy. Pattern fields that are not filled yet are left out in the presentation. In case the pattern authors provided a solution illustration, it is shown as iconic representation in the right part of the header of each pattern which is different from the online library. If no illustration was provided, a generic icon is used for uniformity reasons.

[1]http://pattern-library.sec-bridge.eu

E.1 Laws and Ethical Considerations

The patterns formulated in this hierarchy level should provide the legal backbone of the approaches derived within the project. If important laws or ethical issues are disregarded, the whole system or application concept can be affected. Especially problems like monitoring, data collection and surveillance were often stated during project workshops. Moreover, ethical considerations need to be taken into account in case that official laws might be bended to a certain degree in disaster situations. For example, in a major crisis, it may be of great importance to collect all kinds of information also from private resources in order to get a better picture of the situation. This can include accessing data collected from personal electronic devices owned by bystanders or victims. Regarding the latter, application and technological concepts are subject to research in order to support the search and retrieval process of first responder units. Still, ethical and legal aspects must be discussed and influence development.

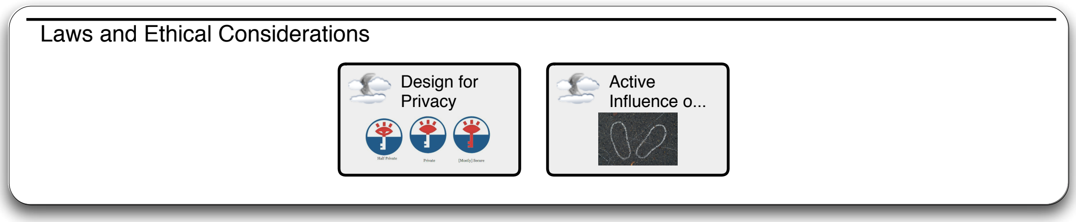
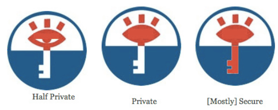


Figure E.1: Available patterns within the hierarchy level "Laws and Ethical Considerations ".

1 Design for Privacy

Under Consideration
Pattern Origin: *Adapted to Project*



Context

Any context where people may process personal data.

Image by digital.democracy http://www.flickr.com/photos/digitaldemocracy

Problem Summary

Privacy is about people's ability to manage boundaries between public and private. It's about informational self-determination. New technologies expand the spatial and temporal reach and persistence of personal data.

Problem Details and Forces

These new opportunities are often great. However, it can also become difficult to managing private and public boundaries. What is great now may be problematic later (e.g. a photo taken during an emergency rescue that shows you're ok to your family, but that also shows you in an incriminating situation).

Solution Summary

Design should extend people's capacity to sense, understand and manage other people's access to their personal data and it should prevent inappropriate and unlawful privacy intrusion.

Comments

- Suggestion for quicker reading the pattern: Splitting up the problem and solution summary to the corresponding details section. Thus, the summaries would only consist of 1-2 sentences and make the reader more "curious" :)
- A very relevant but challenging issue indeed. Specially if the victim is not in a condition to sense, understand, and manage access to their personal data, which is often the case with victims. The role of the public and media is important, but how to "mark" privacy policy on material that is produced in the situation? Like photos of victims?

2 Active Influence on Logging

Under Consideration
Pattern Origin: *Adapted to Project*

Context

Emergency responders already log key decisions and events. Technology is part of a transformation of logging capacity and practices.

Problem Summary

Technologies can log human activities and communications that are entered into systems directly or indirectly. However, technologically augmented logging is also part of a transformation of what logging means and what it is used for.

Problem Details and Forces

Logging can be great as it can save vital time, it can take work off people's shoulders, logs can be locked so no-one can change them, machines don't get tired logging, extensively logged real world events can be used for training. It can be used to apportion blame and punishment in retrospect, which, in turn can affect the way in which people carry out their work in the knowledge that everything will be logged.

Solution Summary

There are no 'solutions' - there are lots of ways of managing both positive and negative consequences. For example by installing forgetting, by logging things without identifying persons, by allowing people to turn logging off or erasing their logs.

Comments

- I would like to suggest some smaller changes: 1. Maybe, the title could be shortened to something like: "Log or Don't Log" or "Human-Managed Log" since otherwise, the pattern name itself is a little too long to be remembered easily.

  2. The solution part says that there are no solutions. However, I think that the examples presented are solutions that could be applied. Maybe they could be listed or a little further described in the section about solution details and consequences when applying them.

  3. The Problem Summary could be split up into the central message. The rest of the section could be moved to the problem details. This would improve readability and support a first quick look through the pattern before reading on.

  What does the author think about that? :)

- I understand this pattern that there are pros (technology can log very efficiently / ”it can take work off people’s shoulders” / etc.) and cons (logging can effect the way people work if they know everything they do is logged) of logging. So the pattern focuses tradeoffs between logging and no logging. What I want to mention is: 1.) I agree that logging has direct impact to technology and systems design. Nevertheless I think that the concept of logging itself should be a pattern on a higher layer. I am not familiar with law and regulations in the field of emergency response but I will give an example from the domain of ERP systems: Logging on business transaction level is regulated by law because a ERP system has to be revision-safe. So perhaps there are also such forces in the field of emergency response that directly lead to a ”Logging Pattern” in the layer ”Processes and Concepts” or ”Domain Practices” which is derived from a pattern in the ”Laws and Regulations”. Then the ”Log-don’t log” pattern could be logically derived from those patterns and would therefore be much better placed in the whole domain context of emergency response.

  2.) Also if there are no such higher level concepts and regulations in the domain of emergency response like mentioned in 1 I still can extract more than one pattern from the problem description: a) logging can be done quite efficiently by technology, so design applications for logging and take off that work from human shoulders b) there is a need for logging at all c) we can extract data for training purposes using some logging mechanisms

  3.) As I can use some sentences from the problem description very easily to formulate solutions I wonder if the problem section isn’t describing solutions but rather than problem(s).

  Nevertheless I think this pattern opens up a couple of new pattern ideas so this pattern repo is quite the best place to formulate those ideas.

- Isn’t this necessary like for example black boxes on planes?

E.2 General Processes and Concepts

This hierarchy level should provide information about abstract processes that are applied within the domain. In conjunction with derived and generally known concepts, the patterns at this level of the hierarchy explain general interaction, system and application design decisions. Without following the advice given at this level, the probability of inappropriate and rejected developments rises.

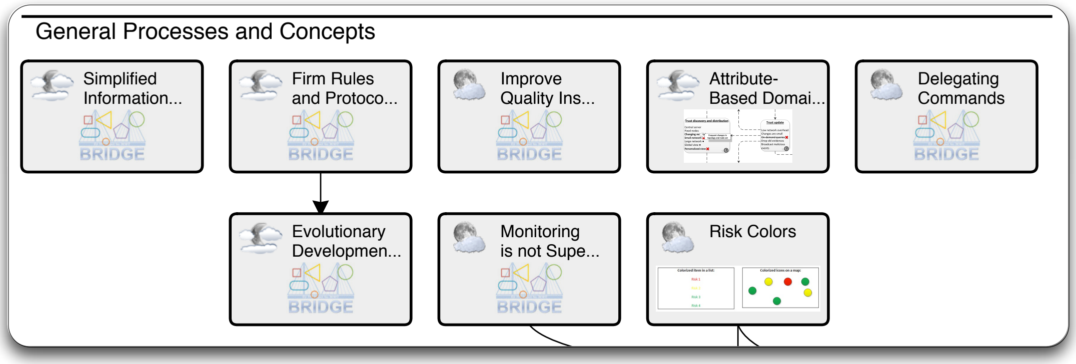


Figure E.2: Available patterns within the hierarchy level "General Processes and Concepts ".

3 Risk Colors

Pattern Candidate
Pattern Origin: *Project-External*

Context

The pattern is relevant for systems that intend to support risk analysis/assessment during emergency response, and particularly in situations with a high number of risks.

Problem Summary

When dealing with risk analysis/assessment, users need an efficient way of separating more critical risks from less critical risks.

Problem Details and Forces

The pattern can only be expected to work when the user of the system are not color blind or severely visually impaired.

Solution Summary

Colorize risk objects (e.g. risk icons on a map, risk items in a list) differently depending on the risk level they represent. Color the most critical risk objects red, less critical risk objects yellow, and the least critical risk objects green.

Supporting Evidence

- *Research Publication:* Research-Based Guidelines for Warning Design and Evaluation (Wolgatera, M. S., Conzolaa, V. C., and Smith-Jackson, T. L.).
  Description: Found at: Applied Ergonomics 33, 3 (2002), 219-230.

Related Patterns

- Triage
- Up-To-Date Vital Values

4 Delegating Commands

Pattern Candidate
Pattern Origin: *Derived from Project*

Context

On-site communication needs to follow strict paths in order to ensure that the chain of command is always informed.

Problem Summary

Crisis response involves diverse stakeholder. Sometimes due to communication problems or hierarchical conditions relevant persons cannot directly communicate with each other.

Problem Details and Forces

The highest commander of a response operation gives a command to a high level commander to forward a specific command to a group residing lower in the hierarchy.

Solution Summary

Maybe it is better to allow for direct notification between command producer and the command receivers, though original intermediaries should still be able to trace or even also get notified about this data transfer to have the common picture.

Supporting Evidence

- *User Workshop:* CTAS Elverum (Norway) Exercise 2011 Video Analysis.
  Description: Attended and evaluated video session within the BRIDGE project.

Comments

- This pattern should be reformulated in a more generic way. The concrete supporting evidence has its own place, where it has already been added.

5 Evolutionary Development

Under Consideration
Pattern Origin: *Project-External*

Context

In the scope of concept, system or device design, new approaches are developed and tried out.

Problem Summary

During design, only a subset of the final requirements is cleared and formulated. The results may not be accepted by the end users.

Solution Summary

Perform evolutionary, iterative design steps with small evaluations in between, e.g. user workshops and demos. Incorporate the feedback that is gathered as quickly as possible.

Supporting Evidence

- *Project Deliverable:* BRIDGE Deliverable 2.1, page 11.

Comments

- Incomplete. Only summaries are provided.

6 Firm Rules and Protocols

Under Consideration
Pattern Origin: *Derived from Project*

Context

Communication design.

Problem Summary

There is a diversity of protocols and communication techniques. Additionally, involved stakeholders apply them differently.

Solution Summary

Use firm rules and communication protocols as scaffolding for flexibility.

Supporting Evidence

- *Project Deliverable:* BRIDGE Deliverable 2.1, page 12.

Related Patterns

- Evolutionary Development

Comments

- Problem details missing but would be needed. The summary alone is too general. Solution details seem to be inappropriate.It seems like it was supposed to be under related patterns.

7 Monitoring is not Supervision

Pattern Candidate
Pattern Origin: *Derived from Project*

Context

Everywhere where new technology introduces monitoring to a process.

Problem Summary

Gathering information by sensors opens the possibility to make this information persistent. It is intended to help users but can be misused for controlling them.

Problem Details and Forces

Monitoring implies the possibility of capturing, combining, and post-processing data, reconfiguring the scope of a process, e.g. like triage, and to include accounting for actions made by responders. Logged information can certainly be helpful for post-operation analysis. However, it should be assured that the detected behavior of responders is only used for process improvement. Knowing about potential other consequences, like dismissal or legal consequences, could negatively influence responders' work in the field.

Solution Summary

Systems should either offer a "forgetting" functionality to delete tracked information after the emergency, or offer the possibility to disable logging if the use of data for reprisal cannot be assured.

Related Patterns

- Triage

Comments

- I would suggest to associate (make a connection in the browse patterns view) to the eTriage pattern as well as the application concept of monitoring vital signs.

  Any comments from others or the author on that?

- Could this also be video material from a surveillance camera, or is it restricted to triage (physiological data? What about marking eTriage as a related pattern?

8 Improve Quality Instead of Speed

Pattern Candidate
Pattern Origin: *Derived from Project*

Context

Derived from triage process in particular but the pattern should be applicable to other processes as well. Processes could be accelerated regarding the time needed to perform them. However, personal presence of first responders helps to calm the victim.

Problem Summary

An accelerated triage process may involve a loss of medical care, or give victims the perception that "nobody is taking real care of them".

Problem Details and Forces

Even if the people who are performing triage usually do not actually provide treatment, victims feel safer if an expert is around and is doing "something". Psychological health also has an influence on the physical wellbeing of patients and panic can be caused if patients are not cared for. A common practice is to connect patients to oxygen bottles although it is not medically indicated, only to calm them down. When a faster triage process reduces the time a responder is close to the patient, the feelings of safety and care can be reduced or even get lost.

Solution Summary

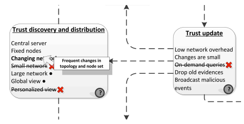
Instead of speeding up the triage process, introduced technology can also aim at improving the process in other ways. For example, the quality of gathered data can be improved.

Comments

- Suggestion to the author: What about a little more general formulation without the hint that the pattern is derived from triage but inverse the argumentation? Thus, this pattern becomes more abstract and general. Then, we ask for associating this abstract concept with concrete examples like triage. Evidence is then given as examples (like triage and others).

9 Attribute-Based Domain Traversal

Under Consideration
Pattern Origin: *Adapted to Project*



Context

Users have to deal with problem domains they are not familiar with. In order to find they way in an unknown domain they can only use the few simplified facts or requirements they know from the current situation. These few facts can be used as attributes to categorize the solutions available.

Problem Summary

In case the user is not familiar with the domain, it may become hard to find or sport available solutions.

Solution Summary

Domains should be divided into categories and existing solutions should be described using these categories. The names of these categories have to be formulated in a way that users can recognize the appropriate category based on the narrow knowledge they have about the current problem.

Supporting Evidence

- *User Workshop:* Workshop on a trust development process.
  Description: The process has been developed in a two stage process. There was an initial requirements workshop followed by a paper prototype being evaluated. The result from the process supports the idea.

Comments

- A bit difficult to grasp the content of this item. Is it meant to support professionals only, and within which emergency agency type?

10 Simplified Information Gathering

Under Consideration
Pattern Origin: *Project-External*

Context

During situation evaluation some contributors of information have no idea what to do and how to follow complex procedures, especially in a stressful environment. Quality of information delivered to decision makers depends for the sake of clarity on the assessment. So clearer questions deliver clever answers in the common case.

Problem Summary

In emergency situations resources are limited and users may need to take tasks they are not familiar with. Information delivery can be affected and can cause problems if misinterpreted.

Problem Details and Forces

There may be situations where users are not able to rely on their own knowledge and experience and they have to take responsibilities for procedures out of their scope. A guided checklist with simple questions structures the user's actions and assessment on site.

Solution Summary

Development of simple and handy yes / no / don't know checklists. Grids of these answers help the professionals to get a better overview of the actual situation and required actions.

E.3 Domain Practices

Within the domain of emergency response, processes are defined or are well established in the every-day practical work. Guidelines and process models that need to be followed by all involved persons as well as established command structures. In addition, custom processes that are not officially documented find their ways into the working environments. Unconscious task further support the successful interaction and cooperation. It is important to know about this kind of tacit knowledge that can only be found during participatory design sessions and via user-centered prototyping.

Figure E.3: Available patterns within the hierarchy level "Domain Practices ".

11 Medical Questionnaire

Pattern Candidate
Pattern Origin: *Project-External*

Context

During during specific phases in disaster situation the health status of injured persons needs to be assessed. An example for such a situation is the triage process.

Problem Summary

The estimation of the health status is performed by medical personal. Sometimes also bystanders or none-medical personal have to give some feedback about the wounded persons in their environment, which is not an easy task.

Solution Summary

Questionnaires can be used to evaluate the health status of a person. Therefore, different questions can be asked to afterwards give estimation about the criticality of the health of this assessed person.

Supporting Evidence

- *Research Publication:* Towards Adaptive User-Interfaces: Developing Mobile User-Interfaces for the Health Care Domain.
  Description: Nestler and Klinker, 2007): Nestler, S. and Klinker, G. - Mobiles Computing in der Medizin (MoCoMed), 2007.

12 Triage

Pattern Candidate
Pattern Origin: *Project-External*

Context

Sorting and prioritization of victims is an essential part of an efficient emergency response during large crises. Triage is the process of sorting injured people into groups based on the severity of their condition, and the need for medical assistance. This prioritization is used in emergency situations when resources are insufficient for all to be treated immediately.

Problem Summary

In accidents with many victims, medical personnel need an easy and immediate way to signal the priority of patients' treatment based on the severity of their condition.

Solution Summary

It is important that everyone immediately understands the prioritization after triage. Thus the tagging must work well on a typical accident scene. The tags must tolerate heat, frost, rain, and must be visible in the darkness. They must also be changeable from one category to another, as triage is often repeated (re-triage).

Supporting Evidence

- *Web Reference:* Wikipedia article on eTriage.
  Description: According to the article, the use of color in triage is a widely accepted standard.
- *Research Publication:* Triage den livsviktige prioriteringen. Rehn M., Vigerust, T., Andersen J.E., Vollebaek L. Ambulanseforum 5:23-26 (2009).

Related Patterns

- eTriage Colors and Icons
- Body Injury Visualization

13 Negotiate Availability

Pattern Candidate
Pattern Origin: *Adapted to Project*

Context

Incident commanders need to coordinate human and material resources. Especially for human resources, the exact state and occupation may be different from the last status update. It is also possible that the current activity covers more than reported.

Problem Summary

Human 'resources' may look available on a screen, but they may not be available in the real situation. If resources that look 'available' are deployed when they are not, in actual fact, available, problems will occur.

Solution Summary

'Deployment' or 'resource allocation' should in most cases involve personal communication.

Supporting Evidence

- *User Workshop:* Stavanger Demo II.
  Description: Summary of a user statement during the prototyping session: If you"re dispatching by dragging that ambulance onto that incident you take away the negotiation. People can't necessarily do that job or not right now or not be the best for it, regardless what is said by the system.

Comments

- 1.) Shouldn't the context of the pattern also mention that a precondition of this pattern is that "availability" or more general "status" of human resources have to be captured some way and also displayed on a screen!?

  2.) I think the part "...but they may not be available in the real situation." in the context section is more a description of the problem rather than a description of the context.

  3.) I think the meaning of the word negotiation in the title goes far beyond just communication as it is mentioned in the solution summary. I will give an example from my own experience from the time I was working as an emergency medical assistant: There is a special kind of radio transceiver in every ambulance. The staff of the car can send their current status to the rescue coordination center just by pressing a button on that radio. Pressing 1 means the status of the ambulance is ”available”, once the rescue coordination center deployed a mission to the ambulance the staff can signal that they are on the way to the place of action pressing 3, pressing 4 they signal we arrived, 7 means we got the patient and drive to the hospital and finally when the patient is at the hospital the staff makes the ambulance ready again and then communicates the status ”available” pressing 1 again. What I want to show with this example is just that there is not just communication but communication following some rules. Especially when the rescue coordination center deploys a job to an ambulance with status ”available” the staff of this ambulance has to acknowledge the job. So that is what I want to mention as negotiation, not just communication. Okay, so what can we derive for the pattern: a) the concept of communication should be elaborated in the solution sections to become much clearer and to match with negotiation in the title. b) in my opinion the concept of negotiation leads directly to some kind of acknowledgment of deployments.

4.) The mentioned findings lead directly to some requirements and therefore patterns in the layer ”User Interface Design” and ”Interaction Design”. Especially I see a connection between the pattern ”Common Interaction” and this pattern since Common Interaction describes ”Enable them to consuming information but also to collaboratively provide and interpret the information.”. Referring this sentence to 3 I feel that we can apply this pattern to achieve ”Common Interaction” at least partly.

5.) The solution illustration should show the negotiation aspect, when I look at the illustration I think of ”planing” and ”decision making” but I can’t see communication and negotiation.

- Addition to 4.) I also think that there is a new pattern in the layer of user interface design which describes defined negotiation possibilities using buttons ore something else to establish the functionality of common interaction and ”Availability is a matter of negotiation”.

14 Separated Components

Under Consideration
Pattern Origin: *Adapted to Project*

Context

Emergency situations often require improvisation due to the lack of time or perfectly appropriate equipment. Malfunctions of the equipment may occur additionally.

Problem Summary

When improvising in an emergency, responders need to know how things can fit together and how they can be taken apart and used for other purposes than intended.

Problem Details and Forces

Example: A radio whose battery dies can still be used with another battery of the same voltage, even if it does not physically fit.

Solution Summary

Design the sub-systems so that it is clear how they connect with each other where the responsibilities of one subsystem end and where those of another begin.

Supporting Evidence

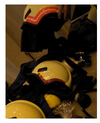
- *Research Publication:* Seamless design in ubiquitous computing (Matthew Chalmers , Ian Maccoll).
  Description: Proceedings of Workshop At the Crossroads: The Interaction of HCI and Systems Issues in UbiComp. 2003

Comments

- Here, I would suggest to split the problem and solution summaries such that the details are shown in the corresponding section. Especially for the solution part :) The summary could provide the essence. The details and consequences further elaborate on the solution. Isn't there a nice iconic picture for this concept? :)
- I suggest changing the name to "Clear Components". Seamful Integration is already used by another pattern and does not really fit here in my opinion.
- Isn't this a problem, as users have to know exactly how things work. Then they must be educated in order to handle the problems, which i find hard to be realized

15 Rigid Structure

Under Consideration
Pattern Origin: *Project-External*



Context

From becoming a firefighter, to commanding an incident, firefighters gradually have to get promoted to make their way up the ranks. A team lead of an engaging team, for instance, is required to have a certain amount of experience. Especially younger firefighters who want to get promoted, reported that this is not easily possible. Only at the higher levels of administration, firefighters are directly appointed senior positions based on their level of formal education.

This pattern is provided by a pattern language derived for a doctoral thesis. The original pattern formulation is available at:

http://patterns.fit.fraunhofer.de/ff/rigid-structure/ A role is assigned before starting the work.

Problem Summary

Firefighting operations face unknown, often chaotic, situations. Nevertheless, firefighters have to act promptly and decisively.

Solution Summary

Therefore, a rigid organizing structure forms the backbone of the operation. Roles are clearly defined and visible, allowing everybody to see who is in charge at different levels. Beyond fixed roles and hierarchies, the structure serves as a means for mutual responsibility and trust.

Comments

- Isn't this already done as it is essential for firefighters to have clear roles?

E.4 Application Concepts

This hierarchy level intends to capture application ideas and knowledge about abstract concepts on how application should work. Existing knowledge should provide a foundation for improving available approaches. Furthermore, it serves as inspiration for new ideas based on the projects goals, user workshops and domain analysis.

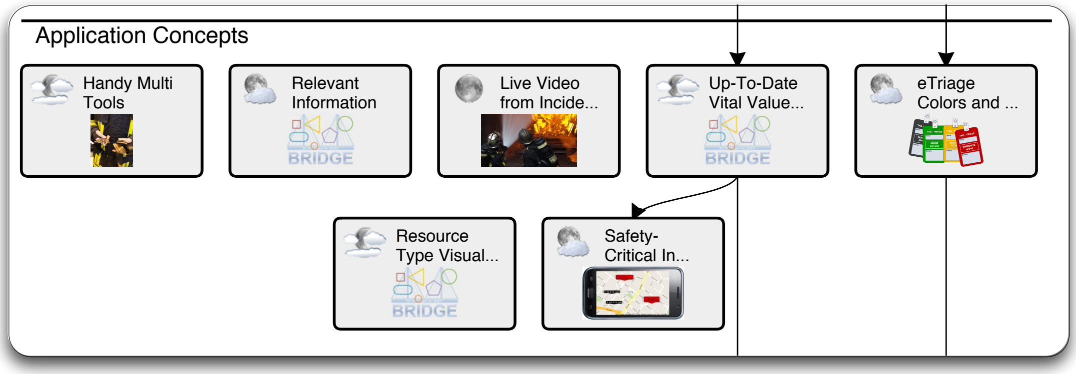


Figure E.4: Available patterns within the hierarchy level "Application Concepts ".

16 Up-To-Date Vital Values

Under Consideration
Pattern Origin: *Adapted to Project*

Context

Vital values of victims need to be monitored and surveilled continuously after care measures.

Problem Summary

Currently, vital values are accessed at time of triage and treatment but sudden decrease of health may not be noticed by medics.

Problem Details and Forces

The vital values of patients are not static. They are changing from time to time. People might be stable at the time when the rescue operation is beginning. But their health can decrease rapidly at any time putting them in danger for life.

Solution Summary

Provide a live overview of the main vital values for every patient and an alarming function when the vital values of a patient come below a threshold.

Related Patterns

- Vital Sign Monitoring
- Safety-Critical Information Display

Comments

- I would split the live overview and alarming function. They may not be both necessary and/or available at the same time.

17 Live Video from Incident

Approved Pattern
Pattern Origin: *Adapted to Project*

Context

Collaboration is crucial for emergency work. Collaboration always means to exchange information. Everybody in the team has to be on the same level of knowledge about an information as good as possible. When a particular team member has an idea about the exact situation of the other member, help can be provided best. This also has to happen from remote locations like the command center.

Problem Summary

It is difficult for the command center to get a feeling about what is currently going on at the emergency side if they only have to rely on irregular updates via radio.

Solution Summary

Providing the command center with a video stream of the view of a responder from the emergency site can enhance the understanding of the current situation in the field.

Supporting Evidence

- *Research Publication:* Visual reporting in time-critical work : Exploring video use in emergency response Chalmers University of Technology. (Bergstrand, Fredrik; Landgren, Jonas).
- *User Workshop:* BRIDGE User Workshop I, Oslo, 29th September 2011. - Blue Sky Group 2: "Patient monitor and health system".
  Description: Video of the eTriage session of the first user workshop at minute 6:30 - Notes eTriage Mockup
- *Research Publication:* Gestures over video streams to support remote collaboration on physical tasks.

Comments

- One important function of the radio updates is that they filter the information to the minimum necessary. Certainly, a lot of detail gets lost, but so does a lot of unworthy information that would only delay the command post.

  The scarcity, or lack of bandwidth, is sometimes a blessing in disguise. While I agree that the command center needs to know what is going on, I don't know whether live video is the right solution for that.

18 Safety-Critical Information Display

Pattern Candidate
Pattern Origin: *Derived from Project*

Context

Doctors treat victims at an emergency site. Additional information about the current patient may be necessary for treatment or safety of the medic due to the victim's background. Other risks may occur that may affect the medical personnel. For example, if an area is not secure for access or if some of the victims are known criminals.

Problem Summary

Information about danger for the medics' lives is often not available to them.

Problem Details and Forces

When doctors are at the emergency site, they focus on getting the vital data about the injured. But they are also in need of information that could hinder their work or even threaten their life. For example, if an area is not secure for access or if some of the victims are known criminals.

Solution Summary

Besides vital data of the patient, additional information about menaces to the medics' life can be visualized on a map in a mobile application which is carried by the medics.

Supporting Evidence

- *User Workshop:* BRIDGE User Workshop I, Oslo, 29th September 2011. Blue Sky Group 3 "Common Information Sharing System".

19 Resource Type Visualization

Under Consideration
Pattern Origin: *Derived from Project*

Context

This pattern is relevant for systems (such as Resource Manager, Akinator) that display resource status. The use of a resource model with pre-defined symbols for different resource types contributes to the situation (resource allocation awareness).

Problem Summary

The selection and prioritization of actions to be executed by mobile workers depends on the availability of information about resource types present in the area.

Problem Details and Forces

Having a standard representation (using different symbols/icons) for different resource types increases the understanding of the situation. This problem needs to be solved both from a central viewpoint, as well as a localized viewpoint.

Solution Summary

1. Use a simple Resource Model as indicated in Figure 1 below. 2. Assign an icon for the main resource types: Human, Information, Vehicle, Equipment, Device, Material, Facility, Infrastructure. 3. Visualize resources on the map by their resource type symbol and their name as annotation (see Figure 2 below).

20 eTriage Colors and Icons

Pattern Candidate
Pattern Origin: *Derived from Project*

Context

New electronic devices are embedded into the triage processes to improve the monitoring of live-values of the patients.

Problem Summary

Users are not used to the electronically augmented approach of existing processes. There may be a problem of interpreting the outputs.

Solution Summary

The colors and icons used for eTriage applications must correspond to the "traditional" triage process.

Designers must ensure the recognition of familiar outputs and colors. This is essential for embedding new technologies into existing processes and tools.

Comments

- Reading the pattern I asked myself if there is really such a thing as a "traditional" triage process all over Europe. While I am sure that there are many similarities and shared patterns (especially with regard to the green-yellow-red color pattern) I wonder in how far triage processes in the different countries differ from each other. If this is the case (and I assume there will be some differences), the pattern could be more clear about what implications this has for the design of triage tools.

21 Relevant Information

Pattern Candidate
Pattern Origin: *Derived from Project*

Context

During an emergency, responders are confronted with many different kinds of information in an unstructured way.

Problem Summary

In emergency cases, first responders need to cope with a lot of information and might get lost.

Solution Summary

Avoid information overload, only provide information relevant for the current user in a current situation.

Also restrict the number of information items per screen.

Supporting Evidence

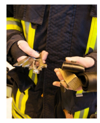
- *User Workshop:* Interview with police officer during Stavanger Demo II.
  Description: Statement from field notes: ”Information overload is an issue right now, despite training, the radio is always too crowded. ”

Comments

- The problem description is too narrow and may lead to misunderstandings. Pattern context should be extended as it assumes a very general applicability which I would challenge.

22 Handy Multi Tools

Under Consideration
Pattern Origin: *Project-External*



Context

Firefighters frequently face problems that require special tools. Physical constraints and time constraints make it impossible to have all the required tools at hand as independent units can neither lift additional load nor have the time and energy to go back to the engine, instead they need to mash-up.

Problem Summary

For a firefighter, there is only so much time for performing on-site actions. Returning to the engine to get additional equipment is usually not an option.

Problem Details and Forces

A standard firefighting mission inside a building on the frontline lasts 30 minutes, the time for which air is available. During that period, the firefighter has to move to the frontline, perform his tasks and return. Usually, this time frame leaves no option to return to an engine during a mission.

With the existing tools, firefighters are heavily equipped.

The standard protection gear of a firefighter weighs about 25 kilograms. It comprises the compressed air cylinder with about 15 kilograms and 10 kilograms for the rest for protective clothing, the helmet, mask, etc.

In addition to that protective gear and dependent on the role that they are assigned, firefighters may carry the nozzle and hoses, a flashlight, an ax, a rope, wooden wedges, wax pens or chalk, a rescue hood for victims, radio, an infrared camera and a variety of small tools such as knives, scissors, screwdrivers, pH paper, oil detection paper, additional plastic gloves for rescue services and gloves for cleaning up after an incident.

This pattern was provided from a pattern collection presented in the author's doctoral thesis. See http://patterns.fit.fraunhofer.de/ff/handy-multi-tools/ for the original pattern description.

Solution Summary

Therefore, firefighters bring tools that can be used for different purposes and invent new ways of using the tools. Tools are designed open for new uses and can be combined with the environment.

E.5 User Interface Design

Patterns on this level deal with smaller parts of applications. They need to take into account device classes, user groups in conjunction with their technical background as well as the standard equipment worn and eventually hindering sight and limiting the exactness of inputs. Both channels, input and output, must be adapted to the current circumstances during which a device and therewith its user interface is applied. Information overload for responding units in the field must be avoided. Important, eventually life-saving, information, however, must be conveyed in a reliable way.

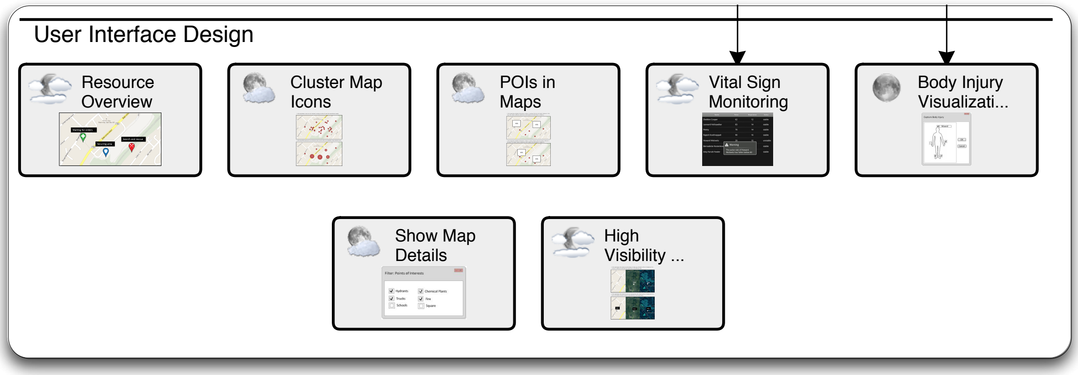
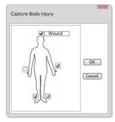


Figure E.5: Available patterns within the hierarchy level "User Interface Design ".

23 Body Injury Visualization

Approved Pattern
Pattern Origin: *Derived from Project*



Context

During search and rescue operations, a victim's injuries need to be documented and transmitted to the hospital for further medical treatment.

Problem Summary

Paramedics need to quickly assess the injuries of a victim during the triage process. The documentation is used by follow-up medical personnel.

Solution Summary

Therefore, show illustrations of the human body according to which injuries can be indicated.

Supporting Evidence

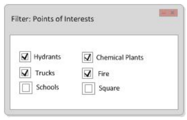
- *Research Publication:* Designing an Emergency Medical Information System for the Early Stages of Disasters in Developing Countries.
  Description: (Sutiono et al., 2010): Sutiono, A.; Qiantori, A.; Prasetio, S.; Santoso, H.; Suwa, H.; Ohta, T.; Hasan, T. andMurni, T. - The Human Interface Advantage, Simplicity and Efficiency Journal of Medical Systems, Springer Netherlands, 2010, 34, 667-675.
- *Research Publication:* ARTEMIS: A Vision for Remote Triage and Emergency Management Information Integration. Dartmouth University. Nov. 2003.
  Description: S. McGrath, E. Grigg, S. Wendelken, G. Blike, M. De Rosa, A. Fiske, and R. Gray
- *Research Publication:* Electronic Triage Tag and Opportunistic Networks in Disasters.
  Description: Martin-Campillo, Yoneki, and Crowcroft - ACM Special Workshop on the Internet and Disasters in CoNext, 2011.

Comments

- The text is a bit shallow, e.g. it is no clear who the ”user” is, who should capture the injuries. That could be a problem if a non-professional should capture injuries because he just doesn’t have the knowledge. On the other hand, a professional might loose important time with capturing injuries. That said, I think the pattern could need some supporting evidence, a problem description and for sure some illustrations.
- This item should obviously be related to other triage patterns
- This is a very clear description.

24 Show Map Details

Pattern Candidate
Pattern Origin: *Project-External*



Context

During a disaster relief teams collect detailed information about the incident site. They use maps to visualize the collected information with the most relevant aspects of the incident.

Problem Summary

The person has to select the points-of-interests that should be visualized in a map.

Problem Details and Forces

Therefore, the user can switch between different layers of detail to show only current relevant information. This should prevent the user from information overload and gives him a focus on specific information as only current important points-of-interests are displayed.

Solution Summary

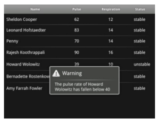
Give the user the possibility to select multiple points-of-interests at one time. This can be done through a dialog, a context menu or a selection before the map or graphic is displayed.

Supporting Evidence

- *Research Publication:* Mobile Computing in Urban Emergency Situations: Improving the Support to Firefighters in the Field Expert Systems with Applications. *Description:* Monares, A. Ochoa, S. F.; Pino, J. A.; Herskovic, V.; Rodriguez-Covili, J. and Neyem - Pergamon Press, Inc., 2011, 38, 1255-1267.

25 Vital Sign Monitoring

Under Consideration
Pattern Origin: *Derived from Project*



Context

The vital values of patients are not static. They are changing from time to time. People might be stable at the time when the rescue operation is beginning. But their health can decrease rapidly at any time putting them in danger for life.

Problem Summary

Currently vital values are accessed at time of triage and treatment but sudden decrease of health may not be noticed by medics.

Solution Summary

Provide a live overview of the main vital values for every patient and an alarming function when the vital values of a patient come below a threshold.

Supporting Evidence

- *User Workshop:* BRIDGE User Workshop I, Oslo, 29th September 2011. - Blue Sky Group 2: "Patient monitor and health system".
  Description: Video of the eTriage session of the first user workshop at minute 6:30 - Notes eTriage Mockup

Comments

- This is a duplicate of "Up-To-Date Vital Values" but I prefer this one for more completeness.
- Perhaps add "continuous" to the heading of this candidate, to highlight the time dimension. And relate it to other triage patterns.
- Since this pattern describes a more concrete way of solving the general problem of continuos vital signs, both patterns should be kept. However, Erion, you are right: The distinction between the two patterns and according fields should be adapted.

26 Cluster Map Icons

Pattern Candidate
Pattern Origin: *Project-External*

Context

In emergency management systems, map-based interfaces might be used to show information about resources, patients, victims, or other points of interest. When these interfaces makes use of markers/icons to represent such points of interest, they will often get cluttered due to the large amounts of markers/icons that are displayed simultaneously, resulting in information overload for the user. Clustering of map-icons can be used to avoid cluttering the map-based interface, and to reduce information overload.

Problem Summary

Map-based interfaces often get cluttered due to a high amount of icons/markers displayed simultaneously.

Solution Summary

Represent similar points of interest that are located close to each other on the map (depending on the zoom level of the map) by one single cluster icon, instead of having one icon for each single point of interest. The clustering of icons should be relative to the current zoom level of the map.

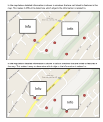
Comments

- This pattern identifies cluttering of visual interfaces with too many individual information 'items' as a problem and proposes clustering as a solution. This is in principle a useful pattern. However, I have a number of questions: * What is the relationship of this pattern to the information visualization literature? There are other ways of aggregating information. I'd like to get a sense of what alternatives would be and for what kinds of information this clustering approach is the best. See e.g. McCandless' work: www.informationisbeautiful.net * What is the basis for 'similar' - simply clustering by number of tweets may or may not be a very useful approach. Who decides what's 'similar', at what point, how can people inspect this? * Generally I am unsure about the pattern structure of 'problem' and 'solution' - it's often more interesting than this and thinking about it in ways other than problem/solution could widen the design space. The problem here could be formulated differently as an opportunity - e.g. 'ability to correlate more data' - dependent on support for human practices of sense-making - design idea:

clustering - design insights: clustering rationale must be inspect able, - related ideas - clustering does not have to be a 'merger' of several small dots into a larger one, it could also be expressed by coloring the map, animating the map, become visible on move-over ...

27 POIs in Maps

Pattern Candidate
Pattern Origin: *Adapted to Project*



Context

In emergency management systems, map-based interfaces might be used to show the location of resources, patients, victims, or other points of interest. When additional information about these points is shown in the interface, it should be clear for the user which information that is related to which point of interest (i.e. detailed information about a given map feature such as an icon should be shown in clear relation to the map feature itself).

Problem Summary

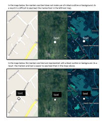
When detailed information about geographical points of interest is displayed in a map-based interface, it can be difficult to determine which point of interest in the map the information is related to.

Solution Summary

Display additional information about geographical points of interest in windows that are visibly linked to the point of interest the information is relevant for.

28 High Visibility of Markers and Text

Under Consideration
Pattern Origin: *Adapted to Project*



Context

In emergency management systems, map-based interfaces might be used to show information about resources, patients, victims, or other points of interest. When these interfaces makes use of markers, icons or text to represent such points of interest, these markers should be clearly visible regardless of the underlying map-type that is being used (e.g. topographic, satellite images, street images).

Problem Summary

The visibility of markers, icons and text used to represent points of interest in map-based interfaces is affected by the color-scheme of the underlying map type. As a result, such markers, icons and text can become harder to see/read when certain map types are being used.

Solution Summary

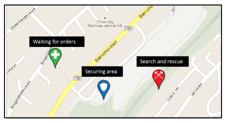
Enclose all icons, markers, and text used to represent points of interest in map-based interfaces with a black background or outline, and colorize the icons, markers and text with bright colors.

Comments

- The solution offered can be improved by not suggesting a color (black), but rather suggesting that the icon/text background should contrast the map background.
- This pattern sounds very obvious! Maintaining high visibility of user interfaces is IMHO not specific to the given context.
- ... either the specificities of the context should be explained, or the context could be formulated in a more general way.

29 Resource Overview

Under Consideration
Pattern Origin: *Derived from Project*



Context

During emergency management, commanding personnel needs to coordinate and distribute the effort of the first responders as efficiently as possible. To achieve this, the commanding personnel needs up-to-date information about the available resources.

Problem Summary

In order to efficiently coordinate and distribute the effort of first responders, commanding personnel needs quick access to information describing the current location and status of these responders.

Solution Summary

Provide commanding personnel with a map-based overview where resources are shown on their current location in the map by means of icons (use different icons to represent different types of resources), and where the current status of the resources are displayed in a label attached to the icon.

E.6 Interaction Design

Considerations on device sizes and handling must be taken into account. Depending on the current situation, equipment and available time to perform tasks, the information bandwidth must be modified. Interaction paths must adapt to situations and not burden the users with additional cognitive loads in stressful time-critical situations.

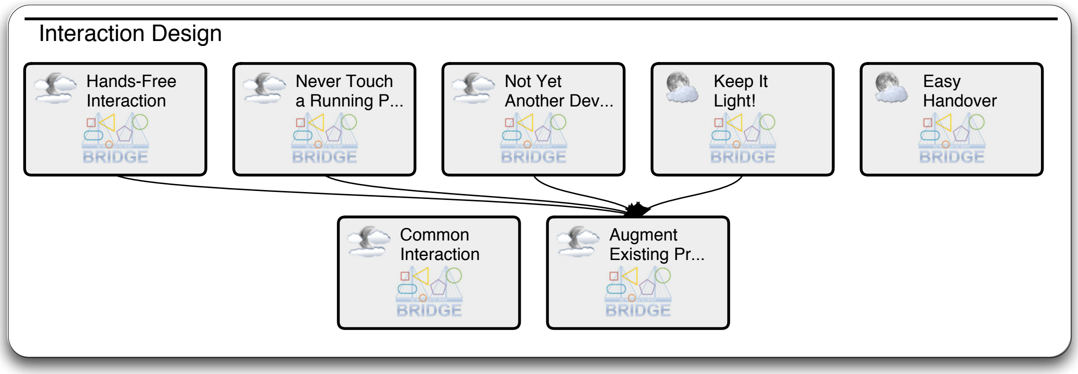


Figure E.6: Available patterns within the hierarchy level "Interaction Design ".

30 Easy Handover

Pattern Candidate
Pattern Origin: *Derived from Project*

Context

You are interacting with a device as somebody else takes over. This person would like to either continue the task you were performing or to start all over easily.

Problem Summary

After taking over a device, the user needs as fast and easy orientation of the device's state. Either the decision is to continue with the current state or an easy way to restart is necessary.

Solution Summary

Design your device with an easy to restart way or an easy interface which has no need of being studied in detail.

31 Keep It Light!

Pattern Candidate
Pattern Origin: *Project-External*

Context

When designing new solutions or augmenting existing ones, additional hardware is often brought in.

Problem Summary

A device's weight adds complexity and slows down helpers.

Solution Summary

Avoid the need for extra devices for your solutions.

Supporting Evidence

- *User Observation:* Observation, Ethnography.
  Description: It is obvious how much weight firefighters already need to carry. Other rescuers also have typically heavy tools to deal with.

Related Patterns

- Augment Existing Processes and Tools

Comments

- The context should give more details about the target group that the authors had in mind for designing new devices. However I support this pattern as in the given context the weight of additional devices is a real issue.

32 Augment Existing Processes and Tools

Under Consideration
Pattern Origin: *Project-External*

Context

When integrating new technologies for supporting tasks, new devices or sensors usually need to be introduced.

Problem Summary

New devices, sensors or other less technical equipment may lead to changing existing tools or even introducing new ones. There is also a danger, that existing processes need to be changed. All this may lead to principle nonacceptance.

Solution Summary

Try to embed the new technology or approach into existing hardware. In a best case scenario, the process or look of the devices do not have to be changed.

Supporting Evidence

- *Presentation:* Jonas Landgren Keynote at BRIDGE TCC meeting in Nov'11 Birlinghoven.
- *User Workshop:* Experienced FF during Stavanger I - BRIDGE Demo II.
  Description: All BRIDGERs who attended sessions with him, could experience a strong reluctance to introduce any new technology.

Comments

- "This will lead to seamless integration the user are unaware of but still are supported."

  If a new technology is seamlessly integrated, but the opportunities, e.g. in terms of communication technology it could be metrics like range, reliability etc, are improved it might be good to give the users a hint of these new opportunities.

33 Common Interaction

Under Consideration
Pattern Origin: *Derived from Project*

Context

At the command post, many persons in charge work together by bringing in new information, organizing and delegating it.

Problem Summary

Many people deal with lots of information that is coming in different intervals and amounts. Everybody wants to share and receive new information as fast as possible.

Solution Summary

Make use of media that can be used by many people at the same time. Enable them to consuming information but also to collaboratively provide and interpret the information.

Comments

- Solution details should be described here not referenced because media that can be used by many people at the same time sounds like a very complex problem.
- This item is related to the master, but is too general. It is not necessarily ”many people” at the control post, and they do not always want to share ”everything” with ”everyone”.

34 Hands-Free Interaction

Under Consideration
Pattern Origin: *Project-External*

Context

First responders carry around a variety of tools and devices within their equipment.

Problem Summary

First responders need to work with different tools and devices during their duty. However, they need to work in parallel: Using the devices, carrying other things or assisting injured people. They need their free hands as often as possible.

Solution Summary

Integrate devices for information within other devices, clothes equipment. Make use of different channels whenever possible (visual, auditive, tactile) depending on the amount of information to be conveyed.

Related Patterns

- Augment Existing Processes and Tools

Comments

- Problem context and summary would deduce a different solution as provided although I see the solution approach as sensible.
- The pattern context/usage text could be expanded. Add illustration; the sentence ”items must quickly be returned to their position” is out of scope, since the solution suggested to integrate devices into other devices. Therefore we can’t influence the ”quickness of returning” that much. Consequences: sensory pollution/overwhelming if the wrong channel is chosen.

35 Not Yet Another Device!

Under Consideration
Pattern Origin: *Derived from Project*

. . . .

Context

New technologies and supportive gadgets are often built within new hardware.

Problem Summary

New hardware adds to the first responders equipment. There is no more space for new devices and the handling of the device has to be learnt.

Problem Details and Forces

The reason for often not accepting new solutions origins from the demand on the user to learn the new device's usage and embed the new functionality to the existing behavior. This may lead to nonacceptance of the device such that it is not taken and used or - if taken - ignored during tasks. The mental load is raised, new training is required since the first responder does not want to be distracted by things he needs to focus on.

Solution Summary

1) Piggyback on an existing process or solution without changing it 2) Piggyback on an existing process or solution with minimal additions to it (augmentation/little training) 3) Piggyback on an existing process or solution with changes (change/training) 4) Introduce a new process or solution with no or little need for training and interaction 5) Introduce a new process or solution

Related Patterns

- Augment Existing Processes and Tools

Comments

- The solution text belongs in the problem summary. The solution could be edited to "When designing a new solution, try the following in order of preference: 1) Piggyback on an existing process or solution without changing it (augmentation/implicit interaction) 2) Piggyback on an existing process or solution with minimal additions to it (augmentation/little training) 3) Piggyback on an existing process or solution with changes (change/training) 4) Introduce a new process or solution with no or little need for training and interaction 5) Introduce a new process or solution

- Agree to the previous comment. Solution Details does not provide solutions, it's rather a problem description. Furthermore, the mental load might also raise due to added functions on existing devices.
- Thank you! I incorporated your suggestions. Please have a look again.

36 Never Touch a Running Process!

Under Consideration
Pattern Origin: *Derived from Project*

Context

New technology is introduced to support the stakeholders. Sometimes, new concepts introduce new processes.

Problem Summary

People are used to their known processes. They might refuse to adopt to a new process that is introduced as concept or as new tool or device.

Solution Summary

If new technology needs to be introduced, try to make their usage as similar to their known processes as possible.

Supporting Evidence

- *Research Publication:* A pattern language of firefighting frontline practice to inform the design of ubiquitous computing (Ph.D Thesis S. Denef).

Related Patterns

- Augment Existing Processes and Tools

Comments

- This pattern seeks to develop sensitivity to people's existing practices (here called 'processes') and advises to mimic existing processes as much as possible in anticipating how a new technology might 'fit' into existing work practices.

  While I really appreciate the intention behind this, and completely agree that sensitivity to existing practices is essential for good design, I also radically disagree with the conclusion. It's much more interesting than this. Just a couple of points to start a discussions:

  1. To think of existing practices as 'processes' is not consistent with the vocabulary I know of collaborative design and ethnographically informed design. Here people talk about 'work practices'. I think the term 'practices' is more useful, because 'processes' assumes a somewhat automatic process (;-)). A distinction can be made between processes and practices that's similar to the distinction between

'plans and situated action' in the literature I know (which represents academic research of 20+ years into IT innovation). Processes and plans are all very well, but actually, people put them into practice and this is a sophisticated, delicately coordinated, often creative and contingent process. So what we really need to understand is practices, not processes. (see e.g. Suchman 2007, Human-machine configurations.

2. Design as preserving existing practices is a highly problematic principle. People's practices have never been static, there are constant transformations of how people do what they do. Design should be about transforming and supporting practice in new ways which generate an overall 'better' situation. This is all very contingent on what people perceive as better, the potential of technology, the constraints of human practices. For example - emergency responders struggle to realize the potential for more interoperable, agile ICT supported crisis response, because there are organizational, social, legal, ethical, political constraints on their everyday practices. Does that mean ICT should support not very interoperable, not very agile practices? Yes and no - design should respect the fact that responders do NOT want to share data with all parties. We should understand why humanitarian organizations don't want to give personal victim data to statutory responders (as in 7/7), and we should understand how they do share data (e.g. via inter-agency liaison officers). We should discuss with them, enable collaborative learning about, and experiment with the potential of ICT to support control over what data is being shared with whom in new ways, ideally with prototypes. This is likely to engender visions for novel ways of doing the work in better ways. If you make the purpose of design to preserve existing practice, you're shooting yourself in the foot. We need to bring about desirable and workable configurations of future practices that incorporate the full potential of new technologies in responsible ways. So the pattern should be 'Respect existing practices, but aim for better futures!'

E.7 Technology and System Design

Technical realizations dealing with system behavior, fault tolerance and quality of information are part of this hierarchy level. The design of system components in terms of hardware and software must take into account the application in harsh environments and therewith mobility issues as well as the robustness of the system. In crisis situations, special affordances and limitations exist on communication channels, network availability and bandwidth.

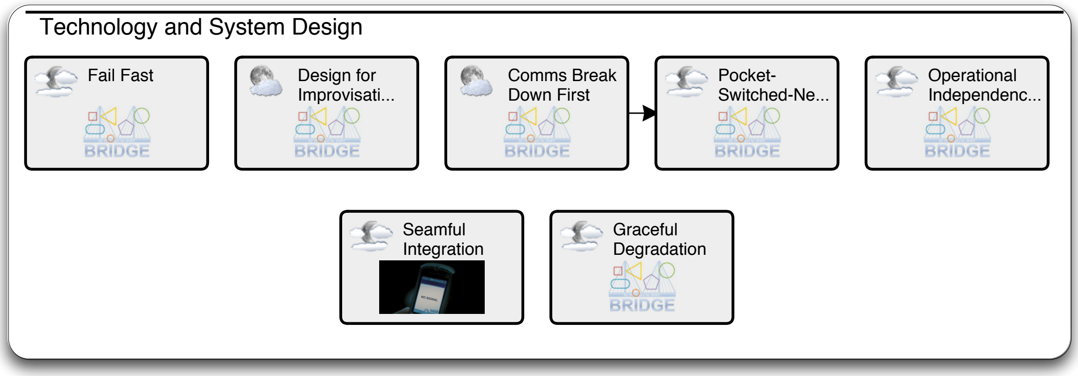


Figure E.7: Available patterns within the hierarchy level ”Technology and System Design ”.

37 Graceful Degradation

Under Consideration
Pattern Origin: *Project-External*

Context

During an emergency when fast response and reaction to the current situation is essential.

Problem Summary

In case that a device falls out during an emergency, there are no time and no resources to repair it.

Solution Summary

Build your system so that it degrades gracefully.

Supporting Evidence

- *User Workshop:* Interview at German technical relief in Bonn.
  Description: Summary of a statement from an interviewee: "...the process needs to go on. If all technology fails you go back to stone age. That means in worst case we work with pencil and papers and send messages via mediators.(?) In large-scale disasters in the beginning frequent breakdowns of technology are the norm.

Refuting Evidence

- *Existing Prototype:* Help Beacons concept and prototype.
  Description: If the bandwidth between two devices is strong enough details are interchanged, if not the seeking device just connects to the device in beacon mode.

Comments

- Title: I'd recommend to put at least in brackets or inside a description field (aka fault-tolerance). See also Wikipedia on graceful degradation.

  The usage is not well expressed. IMHO, better would be to say: the process of emergency response needs to continue regardless of any partial breakdowns.

  Solution: Don't constrain it two a device. It terms of software it could only be a feature/functionality. To be abstract I'd recommend to write sub-system instead of device.

38 Comms Break Down First

Pattern Candidate
Pattern Origin: *Project-External*

Context

New technologies allow extensive sensor data collections and monitoring.

Problem Summary

During the design of supportive technologies for first responders, network connection plays a very important rule for the concepts. However, a daily network that is usually present before a disaster usually breaks down as first. This way, the networks we are usually used to are not present anymore.

Problem Details and Forces

During large-scale emergencies, communication in public networked may quickly become jammed by the users. As a consequence, communication breaks down quickly after the incident.

There is also an additional danger that the infrastructure for the communication network also becomes destroyed or is severely damaged.

Solution Summary

Do not design for large scale, high bandwidth and public network infrastructure.

Build your own devices to work autonomously without a pre-existing network.

Without ever seeing a network, take the most advantage of intermittent connections.

Supporting Evidence

- *Book:* TELECOMMUNICATIONS INFRASTRUCTURE IN DISASTERS: Preparing Cities for Crisis Communications (Anthony M. Townsend Mitchell L. Moss).

Related Patterns

- Pocket-Switched-Network

39 Design for Improvisation

Pattern Candidate
Pattern Origin: *Derived from Project*

Context

Prototypical design for applications and devices.

Problem Summary

Especially within the prototyping and conceptualization phase, systems or devices are constructed focusing on few certain aspects. When extending the concept, the available prototype cannot be used anymore.

Solution Summary

Design for improvisation. Think about clear structure, architecture. Provide interfaces and make use of extensible soft- and hardware components.

Comments

- I think this pattern is not very clear. What does it mean to design for improvisation? Isn't a clear structure and architecture always good in technology design? I think this pattern depends strongly on what you want to do with the prototype, and the phase of the development you are in. In early stages, there could be a different solution to the problem that is presented here: design *lightweight* prototypes that can be thrown away if they don't work, test concepts first with demonstrators/mock-ups that don't take so much time to prepare ... or simply put, follow an agile way of technology development. In the long term, the mentioned principles make sense, but I think in this pattern we should differentiate a bit.

40 Operational Independence

Under Consideration

Pattern Origin: *Derived from Project*

Context

System / Concept Design for usage in the field.

Problem Summary

Many devices and systems use and share data. Often, hard coupling happens during system design.

Problem Details and Forces

Many devices and systems use and share data. Often, hard coupling creeps in during system design and creates conditions for strong dependencies between devices and systems, such that when one fails, the others must necessarily fail.

Solution Summary

Keep as a design guideline that loose coupling between the system and its devices and among the devices must be kept in mind during design.

Supporting Evidence

- *Project Deliverable:* BRIDGE Deliverable 2.1, page 11.

Comments

- Context/Usage does not tell me much. Other fields should be filled too.

  I understand what the author wants to say, but I would express it a bit longer, e.g. "Many devices and systems use and share data. Often, hard coupling creeps in during system design and creates conditions for strong dependencies between devices and systems, such that when one fails, the others must necessarily fail.

41 Seamful Integration

Under Consideration
Pattern Origin: *Adapted to Project*

Context

When designing ubiquitous computing solutions several technologies at different levels of maturity may be used.

Problem Summary

Hiding the inner workings of infrastructure often prevent people to find solutions to overcome problems with the usage of technology.

Solution Summary

Allow people to have a glimpse of the inner workings of the technologies involved, to allow for creative recomposition of technological elements.

42 Fail Fast

Under Consideration
Pattern Origin: *Adapted to Project*

Context

Emergency response is time-critical.

Problem Summary

Establishing network connections or downloading something can fail.

Solution Summary

Failing should happen fast to save people from wasting time, allowing them to find alternatives.

Comments

- Actually, I think that the second part of the problem could be formulated as solution for system design. This must be kept in the back of the designer's head. Don't let the people involved in the emergency waste time. Great abstract concept! :)
- Could be rephrased as the need for "continuous status information" - in the cases where the outcome is (still) unknown. If access to a network is not possible, you would appreciate the system to retry later.

43 Pocket-Switched-Network

Under Consideration
Pattern Origin: *Adapted to Project*

Context

On a disaster field, public communication networks are usually overloaded or broken down.

Problem Summary

It is difficult, if not impossible, to use them for data exchange by responders. Moreover, creating an own network is not very easy, and there is always a chance that this ad-hoc network will not function at some point or for some area.

Problem Details and Forces

Networks go down or are disrupted, and it is not easy to create *reliable* ad-hoc networks for emergency workers. At the same time it is necessary to send and exchange data, especially with the command post.

Solution Summary

Minimize the amount of data transfer needed, perhaps by using efficient formats instead of bloated ones.

Supporting Evidence

- *Research Publication:* Pocket switched networks and human mobility in conference environments.
- *Research Publication:* Using Haggle to create an electronic triage tag.
- *Research Publication:* Electronic Triage Tag and Opportunistic Networks in Disasters.
  Description: Martin-Campillo, Yoneki, and Crowcroft - ACM Special Workshop on the Internet and Disasters in CoNext, 2011.

Comments

- I suggest splitting up the summaries and swapping out the details to the corresponding details sections.

Bibliography

Pablo Acuna, Paloma Diaz, and Ignacio Aedo. Development of a Design Patterns Catalog for Web-based Emergency Management Systems. In *Proceedings of the 7th International ISCRAM Conference*. ISCRAM, 2010.

Christopher Alexander. *Notes on the Synthesis of Form*. Havard University Press, Cambridge, Massachusetts, 7 (2002) edition, 1964.

Christopher Alexander. *A Pattern Language: Towns, Buildings, Construction*. Oxford University Press, New York, NY, USA, 1977.

Christopher Alexander. *The Timeless Way of Building*. Oxford University Press, New York, New York, USA, 1979.

Christopher Alexander, Sara Ishikawa, and Murray Silverstein. *A Pattern Language which Generates Multiservice Centers*. Center for Environmental Structure, Berkeley, California, 1968.

William L. Anderson and William T. Crocca. Experiences in Reflective Engineering Practice: Co-Development of Product Prototypes. In *Proceedings of the Participatory Design Conference*, pages 13–22, Cambridge, Massachusetts, November 1992. Computer Professionals for Social Responsibility.

N. Arranz and J. C. Fedez de Arroyabe. Complex joint R&D projects: From empirical evidence to managerial implications. *Complexity*, 15(1):61–70, 2009.

Anna Averbakh, Eric Knauss, and Olga Liskin. An Experience Base with Rights Management for Global Software Engineering. In *Proceedings of the 11th International Conference on Knowledge Management and Knowledge Technologies*, i-KNOW '11, New York, NY, USA, 2011. ACM. doi: 10.1145/2024288.2024300.

Thomas Bader, Andreas Meissner, and Rolf Tscherney. Digital Map Table with Fovea-Tablett®: Smart Furniture for Emergency Operation Centers. In *Proceedings of the 5th International Conference on Information Systems for Crisis Response and Management ISCRAM*, pages 679–688, Washington DC, 2008.

Sidney Bailin. Software Development as Knowledge Creation. *International Journal of Applied Software Technology*, 3(1):75–89, 1997.

Elisabeth Bayle, Rachel Bellamy, George Casaday, Thomas Erickson, Sally Fincher, Beki Grinter, Ben Gross, Diane Lehder, Hans Marmolin, Brian Moore, Colin Potts, Grant Skousen, and John Thomas. Putting It All Together: Towards a Pattern Language for Interaction Design: A CHI 97 Workshop. *SIGCHI Bull.*, 30(1):17–23, 1998. doi: 10.1145/280571.280580.

Kent Beck, Ron Crocker, Gerard Meszaros, James O. Coplien, Lutz Dominick, Frances Paulisch, and John Vlissides. Industrial Experience with Design Patterns. In *Proceedings of the 18th International Conference on Software Engineering*, pages 103–114, 1996. doi: 10.1109/ICSE.1996.493406.

Joseph Bergin. The Pedagogical Patterns Project. Available online: http://www.pedagogicalpatterns.org [Accessed on 26 September 2013].

Joseph Bergin, Jutta Eckstein, Markus Völter, Marianna Sipos, Eugene Wallingford, Klaus Marquardt, Jane Chandler, Helen Sharp, and Mary Lynn Manns. *Pedagogical Patterns: Advice For Educators*. CreateSpace Independent Publishing Platform, 2012.

Fredrik Bergstrand and Jonas Landgren. International Journal of Emergency Management. *Information Sharing Using Live Video in Emergency Response Work*, 6(3-4):295–301, 2009.

Hugh Beyer and Karen Holtzblatt. *Contextual Design*. Morgan Kaufmann, 1998.

Barry W. Boehm. A Spiral Model of Software Development and Enhancement. *SIGSOFT Softw. Eng. Notes*, 11(4):14–24, August 1986. doi: 10.1145/12944.12948.

Jan Borchers. *A Pattern Approach to Interaction Design*. John Wiley & Sons, West Sussex, England, 1st edition, 2001.

Marion Buchenau and Jane Fulton Suri. Experience Prototyping. In *Proceedings of the 3rd Conference on Designing Interactive Systems: Processes, Practices, Methods, and Techniques*, DIS '00, pages 424–433, New York, NY, USA, 2000. ACM Press. doi: 10.1145/347642.347802.

Frank Buschmann, Regine Meunier, Hans Rohnert, Peter Sommerlad, and Michael Stal. *Pattern-Oriented Software Architecture, Volume 1: A System of Patterns*. Wiley, Chichester, UK, 1996.

Bill Buxton. *Sketching User Experiences: Getting the Design Right and the Right Design*. Morgan Kaufmann, 2007.

William Buxton and R. Sniderman. Iteration in the Design of the Human-Computer Interface. In *Proceedings of the 13th Annual Meeting of the Human Factors Association of Canada*, pages 72–81. HFAC, 1980.

Liz Carver and Murray Turoff. Human-Computer Interaction: The Human and Computer as a Team in Emergency Management Information Systems. *Communications of the ACM*, 50(3):33–38, 2007. doi: 10.1145/1226736.1226761.

Simon Connelly, Jay Burmeister, Anthony MacDonald, An Hussey, and Drew. Extending and evaluating a pattern language for safety-critical user interfaces. *Conferences in Reseach and Practice in Information Technology*, 3, 2001.

James O. Coplien and Neil B. Harrison. *Organizational Patterns of Agile Software Development*. Pearson Prentice Hall, Hamilton, NY, USA, 2005.

A. Cornils and G. Hedin. Tool Support for Design Patterns Based on Reference Attribute Grammars. In *Proceedings of WAGA'00*, Ponte de Lima, Portugal, 2000.

Nils Dahlbäck, Arne Jönsson, and Lars Ahrenberg. Wizard of Oz Studies: Why and How. In *Proceedings of the 1st International Conference on Intelligent User Interfaces*, IUI '93, pages 193–200, New York, NY, USA, 1993. ACM. doi: 10.1145/169891.169968.

Edward De Bono. *De Bono's Thinking Course*. BBC Active, Harlow, GB, 1st new edition, 2006.

Lotte de Rore. *Measuring Productivity and Improving Efficiency in Software Development Environments.* Phd thesis, Katolieke Universiteit Leuven, 2009.

Sebastian Denef. *A Pattern Language of Firefighting Frontline Practice to Inform the Design of Ubiquitous Computing*. PhD thesis, TU Delft, Netherlands, RWTH Aachen Unversity, Fraunhofer FIT, 2011.

Junhua Deng, Elizabeth Kemp, and E G Todd. Managing UI Pattern Collections. In *Proceedings of the 6th ACM SIGCHI New Zealand Chapter's International Conference on Computer-Human Interaction: Making CHI Natural*, CHINZ '05, pages 31–38, New York, NY, USA, 2005. ACM. doi: 10.1145/1073943.1073951.

Klaus Didrich and Torsten Klein. A Pragmatic Approach to Software Documentation. In *Forschungsberichte des Fachbereichs Informatik*, volume 96. Berlin, Germany, 1996.

Alan Dix, Janet Finlay, Gregory Abowd, and Russel Beale. *Human-Computer Interaction*. Prentice Hall Inc., Upper Saddle River, NJ, USA, 2nd edition, 1998.

Steven P. Dow, Alana Glassco, Jonathan Kass, Melissa Schwarz, Daniel L. Schwartz, and Scott R. Klemmer. Parallel Prototyping Leads to Better Design Results, More Divergence, and Increased Self-Efficacy. *ACM Transactions on Computer-Human Interaction*, 17(4):1–24, December 2010. doi: 10.1145/1879831.1879836.

Gilles Dubochet, C Exton, and J Buckley. Computer Code as a Medium for Human Communication: Are Programming Languages Improving? In *21st Annual Workshop of the Psychology of Programming Interest Group*, 2009.

Amnon H Eden, Amiram Yehudai, and J Gil. Precise Specification and Automatic Application of Design Patterns. In *12th IEEE International Conference on Automated Software Engineering (ASE'97) (formerly: KBSE)*, 1997.

Thomas Erickson. Lingua Francas for Design: Sacred Places and Pattern Languages. In *Proceedings of the 3rd Conference on Designing Interactive Systems: Processes, Practices,* *Methods, and Techniques*, DIS ’00, pages 357–368, New York, NY, USA, 2000. ACM. doi: 10.1145/347642.347794.

Michael Falkenthal, Dierk Jugel, René Reiners, Wilfried Reimann, and Michael Pretz. Maturity Assessments of Service-oriented Enterprise Architectures with Iterative Pattern Refinement. In *Informatik 2012 : Was bewegt uns in der/die Zukunft?: Beiträge der 42. Jahrestagung der Gesellschaft für Informatik e.V. (GI), 16. - 12.9.2012 in Braunschweig*, pages 1095–1101, Bonn, Germany, 2012. Gesellschaft für Informatik (GI).

Christoph Fehling, Frank Leymann, Ralph Retter, Walter Schupeck, and Peter Arbitter. *Cloud Computing Patterns: Fundamentals to Design, Build, and Manage Cloud Applications*. Springer, 2013.

Siska Fitrianie and Rothkrantz. Leon. A Visual Communication Language for Crisis Management. *The International Journal of Intelligent Control and Systems*, pages 208–216, 2007.

Christine Floyd, Wolf-Michael Mehl, Fanny-Michaela Resin, Gerhard Schmidt, and Gregor Wolf. Out of Scandinavia: Alternative Approaches to Software Design and System Development. *Human–Computer Interaction*, 4(4):253–350, 1989. doi: 10.1207/s15327051hci0404_1.

Marcus Fontoura and Carlos Lucena. Extending UML to Improve the Representation of Design Patterns. *Journal of Object-Oriented Programming*, 13:12–19, 2001.

Ashraf Gaffar, D. Sinnig, H. Javahery, and Ahmed Seffah. MOUDIL: A Comprehensive Framework for Disseminating and Sharing HCI patterns. In *CHI 2003 Workshop on HCI Patterns: Concepts and Tools*, 2003.

Ashraf Gaffar, Ahmed Seffah, and John A. Van der Poll. HCI Pattern Semantics in XML: a Pragmatic Approach. *SIGSOFT Softw. Eng. Notes*, 30(4):1–7, 2005. doi: 10.1145/1082983.1083112.

Erich Gamma, Richard Helm, Ralph E. Johnson, and John Vlissides. *Design Patterns. Elements of Reusable Object-Oriented Software*. Addison-Wesley Longman, Amsterdam, 1994.

Harold Garfinkel. *Studies in Ethnomethodology*. Prentice Hall Inc., Englewood Cliffs, New Jersey, US, 1967.

Joseph A. Goguen and Charlotte Linde. Techniques for Requirements Elicitation. *Requirements Engineering*, pages 152–164, 1993. doi: 10.1109/ISRE.1993.324822.

John D. Gould and Clayton Lewis. Designing for Usability: Key Principles and What Designers Think. *Communications of the ACM*, 28(3):300–311, 1985.

Ian Graham. *A Pattern Language for Web Usability*. Addison-Wesley, London, UK, 2003.

Joan Greenbaum and Morten Kyng, editors. *Design at Work: Cooperative Design of Computer Systems*. L. Erlbaum Associates Inc., Hillsdale, NJ, USA, 1991.

Thomas Grill and Margit Blauhut. Design Patterns Applied in a User Interface Design (UID) Process for Safety Critical Environments (SCEs). In Andreas Holzinger, editor, *HCI and Usability for Education and Work*, volume 5298 of *Lecture Notes in Computer Science*, pages 459–474. Springer Berlin / Heidelberg, 2008. doi: 10.1007/978-3-540-89350-9_32.

Timothy Hanratty, John Yen, Michael McNeese, Sooyoung Oh, Hyun-Woo Kim, Dev Minotra, Laura Strater, Haydee Cuevas, Dan Colombo, and R J Hammell. Knowledge Visualization to Enhance Human-Agent Situation Awareness Within a Computational Recognition-Primed Decision System. In *Military Communications Conference, 2009. MILCOM 2009. IEEE*, pages 1–7, 2009. doi: 10.1109/MILCOM.2009.5379847.

Wilfred J. Hansen. User engineering principles for interactive systems. In *Proceedings of the Fall Joint Computer Conference (AFIPS '71)*, pages 523–532, New York, NY, USA, 1971. IEEE Computer Society.

John A Hughes, Dave Randall, and Dan Shapiro. From Ethnographic Record to System Design: Some Experiences from the Field. *Computer Supported Cooperative Work*, 1: 123–141, 1992.

Shah Humayoun, Tiziana Catarci, Massimiliano de Leoni, Andrea Marrella, Massimo Mecella, Manfred Bortenschlager, and Renate Steinmann. The WORKPAD User Interface and Methodology: Developing Smart and Effective Mobile Applications for Emergency Operators. In *Universal Access in Human-Computer Interaction. Applications and Services*, volume 5616 of *Lecture Notes in Computer Science*, pages 343–352. Springer Berlin / Heidelberg, 2009. doi: 10.1007/978-3-642-02713-0_36.

Andrew Hussey. Patterns for safety and usability in human-computer interfaces. Technical report, Software Verification Research Centre, The University of Queensland, Queensland, Australia, 1999.

Takashi Iba and Taichi Isaku. Holistic Pattern-Mining Patterns. In *Proceedings of the 19th Conference on Pattern Language of (scheduled for 2014)*. ACM Digital Library, 2014.

Infragistics. The Quince Pattern Library. Available online: http://quince.infragistics.com [Accessed on 26 September 2013].

Project Management Institute. *A Guide To The Project Management Body Of Knowledge (PMBOK Guides)*. Project Management Institute, Newtown Square, Pennsylvania, USA, 4th edition, 2008.

Noriaki Kano, Shin Tsuji, Nobuhiko Seraku, and Fumio Takahashi. Attractive Quality and Must-Be Quality. *Journal of the Japanese Society for Quality Control*, 14(2):39–44, 1984.

Clare-Marie Karat. Cost-benefit analysis of iterative usability testing. In *INTERACT '90: Proceedings of the IFIP TC13 Third International Conference on Human-Computer Interac-*

tion, pages 351–356, Amsterdam, The Netherlands, 1990. North-Holland Publishing Co.

Finn Kensing and Jeanette Blomberg. Participatory Design: Issues and Concerns. *Computer Supported Cooperative Work (CSCW)*, 7(3):167–185, 1998. doi: 10.1023/A:1008689307411.

Deepak Khazanchi, John D. Murphy, and Stacie Petter. Guidelines for Evaluating Patterns in the IS Domain. In *MWAIS 2008 Proceedings*. Midwest Association for Information Systems, AIS Electronic Library (AISeL), 2008.

Dae-Kyoo Kim, Robert France, Sudipto Ghosh, and Eunjee Song. A UML-Based Metamodeling Language to Specify Design Patterns. In *Proceedings of the Workshop Software Model Eng. (WiSME) with Unified Modeling Language Conf. 2003*, 2003.

Christian Kruschitz and Martin Hitz. Are Human-Computer Interaction Design patterns Really Used? In *Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries*, NordiCHI '10, pages 711–714, New York, NY, USA, 2010. ACM. doi: 10.1145/1868914.1869011.

Tibor Kunert. *User-Centered Interaction Design Patterns for Interactive Digital Television Applications*. Springer, London, UK, 2009.

Janne Lammi, Matti Varjokallio, and Johannes Hocksell. Patternry. Available online: http://patternry.com [Accessed on 26 September 2013].

Matt Leacock, Erin Malone, and Chanel Wheeler. Implementing a Pattern Library in the Real World: A Yahoo! Case Study. Available at: www.leacock.com/patterns [Accessed on 26 September 2013], 2005.

Clayton Lewis, Peter G Polson, Cathleen Wharton, and John Rieman. Testing a Walkthrough Methodology for Theory-Based Design of Walk-Up-and-Use Interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '90, pages 235–242, New York, NY, USA, 1990. ACM. doi: 10.1145/97243.97279.

Neil Maiden and Alexis Gizikis. Where Do Requirements Come From? *IEEE Software*, 18:10–12, 2001. doi: 10.1109/52.951486.

Mary Lynn Manns and Linda Rising. *Fearless Change: Patterns for Introducing New Ideas: Introducing Patterns into Organizations*. Addison-Wesley, Boston, MA, USA, 2005.

Gerard Meszaros and Jim Doble. A Pattern Language for Pattern Writing. In Robert C Martin, Dirk Riehle, and Frank Buschmann, editors, *Pattern Languages of Program Design 3*, chapter A pattern, pages 529–574. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1997.

Sebastian Meyer, Anna Averbakh, Torsten Ronneberger, and Kurt Schneider. Experiences from Establishing Knowledge Management in a Joint Research Project. In

Product Focused Software Process Improvement - Lecture Notes in Computer Sciences, volume 7343 of *Lecture Notes in Computer Science*, pages 233–247, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. doi: 10.1007/978-3-642-31063-8_18.

Álvaro Monares, Sergio F Ochoa, José A Pino, Valeria Herskovic, Juan Rodriguez-Covili, and Andrés Neyem. Mobile computing in urban emergency situations: Improving the support to firefighters in the field. *Expert Systems with Applications*, 38(2): 1255–1267, 2011. doi: 10.1016/j.eswa.2010.05.018.

Susana Montero, Paloma Díaz, and Ignacio Aedo. A Semantic Representation for Domain-Specific Patterns. In *International Symposium on Metainformatics*, pages 129–140. Springer-Verlag, 2005.

Michael J. Muller, Daniel M. Wildman, and Ellan A. White. Participatory Design. *Communications of the ACM*, 36(4):24–28, 1993.

Carl Murchison. *Psychologies of Nineteen Thirty (Classics in Psychology Ser.)*. Ayer Co Pub, reprint edition, 1930.

Simon Nestler and Gudrun Klinker. Towards adaptive user-interfaces: Developing mobile user-interfaces for the health care domain. In *Mobiles Computing in der Medizin (MoCoMed)*, Augsburg, Germany, September 2007.

Lindy Newlove-Eriksson and Helena Hermansson. Symbols, Symbology and Systems: A Comprehensive Overview. Technical report, European Commission, 2010.

Jakob Nielsen. Iterative User-Interface Design. *Computer*, 26(11):32–41, 1993. doi: 10.1109/2.241424.

Jakob Nielsen and Jan Maurits Fabe. Improving System usability Through Parallel Design. *IEEE Computer*, 29(2):29–35, 1996.

Erik G. Nilsson. Design patterns for user interface for mobile applications. *Advances in Engineering Software*, 40(12):1318–1328, 2009.

Donald A. Norman. *The Design of Everyday Things*. Perseus Books, 1988.

PatternTap LLC. Pattern Tap. Available online: http://patterntap.com [Accessed on 26 September 2013].

Luka Pavlič, Marjan Heričko, Vili Podgorelec, and Ivan Rozman. Improving Design Pattern Adoption with an Ontology-Based Repository. *Informatica*, 33:189–197, 2009.

Christian R. Prause and René Reiners. A Software Process Framework for PLE Development to Reduce Time and Costs of Widget Creation and Evolution. In *PLE Conference*, number 1988, page 5 pp., 2011.

Christian R. Prause, René Reiners, Silviya Dencheva, and Andreas Zimmermann. Incentives for Motivating Scientists to Develop High-Quality Source Code. In *Pre-*

ceedings of the Psychology of Programming Interest Group (PPIG) Workshop, page 6 pp., Dundee, UK, 2010a.

Christian Reinhard Prause. *Improving the Internal Quality of Software through Reputation-based Gamification*. Doctoral thesis, RWTH Aachen University, Aachen, 2013.

Christian Reinhard Prause, Marius Scholten, Andreas Zimmermann, René Reiners, and Markus Eisenhauer. Managing the Iterative Requirements Process in a Multi-National Project using an Issue Tracker. In *In 3rd International Conference on Global Software Engineering. IEEE*, pages 151–159, Bangalore, India, August 2008. IEEE Computer Society Press.

Christian Reinhard Prause, René Reiners, and Silviya Dencheva. Empirical Study of Tool Support in Highly Distributed Research Projects. In *Preceedings of the 2010 International Conference on Global Software Engineering (ICGSE)*, pages 23 – 32, Princeton / NJ, USA, August 2010b.

René Reiners. Towards a Common Pattern Language for Ubicomp Application Design - A Classification Scheme for Ubiquitous Computing Environments -. In *PATTERNS 2010, The Second International Conferences on Pervasive Patterns and Applications*, pages 28–33. IARIA Conference, Think Mind(TM) Digital Library, November 2010.

René Reiners. An Extended Pattern Language Approach for UbiComp Application Design. In *Lecture Notes in Informatics - Informatiktage 2011*, page 4, Bonn, Germany, 2011a. GI - Gesellschaft der Informatik.

René Reiners. New Pattern Language Concepts for Designing UbiComp Applications Connecting to Cloud Services. In René Reiners and Alfred Zimmermann, editors, *Informatik 2011 : Informatik schafft Communities: Beiträge der 41. Jahrestagung der Gesellschaft für Informatik e.V. (GI), 4. - 7.10.2011 in Berlin*, pages 100–105, Bonn, Germany, September 2011b. GI - Gesellschaft für Informatik.

René Reiners. A Pattern Evolution Process - From Ideas to Patterns. In *Lecture Notes in Informatics Informatiktage 2012*, pages 115–118, Bonn, Germany, 2012. Gesellschaft für Informatik e.V., GI - Gesellschaft für Informatik.

René Reiners, Irina Astrova, and Alfred Zimmermann. Introducing new Pattern Language Concepts and an Extended Pattern Structure for Ubiquitous Computing Application Design Support. In René Reiners and Alfred Zimmermann, editors, *PATTERNS 2011, Third International Conferences on Pervasive Patterns and Applications*, pages 61–66. XPS - Xpert Publishing Services, September 2011.

René Reiners, Michael Falkenthal, Dierk Jugel, and Alfred Zimmermann. Requirements for a Collaborative Formulation Process of Evolutionary Patterns. In *preparation to appear in the Proceedings of the 18th European Conference on Pattern Languages of Programs EuroPLoP '13 (scheduled for 2014)*. ACM Digital Library, 2014a.

René Reiners, Ragnhild Halvorsrud, Aslak Wegner Eide, and Daniela Pohl. An Approach to Evolutionary Design Pattern Engineering. In *Proceedings of the 19th international Conference on Pattern Languages of Programs PLoP 2012 (scheduled for 2014)*. ACM Digital Library, 2014b.

Elena Revilla, Joseph Sarkis, and Juan Acosta. Towards a knowledge management and learning taxonomy for research joint ventures. *Technovation*, 25(11):1307–1316, 2005. doi: 10.1016/j.technovation.2004.06.005.

Bernd Rohrbach. Kreativ nach Regeln - Methode 635. *Absatzwirtschaft*, 12(19):73/76, 1969.

Jean-Marc Rosengard and Marian F. Ursu. Ontological Representations of Software Patterns. In *Lecture Notes in Computer Science*, volume 3215, pages 31–37. Springer Berlin / Heidelberg, 2004.

Winston W. Royce. Managing the Development of large Software Systems: Concepts and Techniques. *Proceedings of the 9th International Conference on Software Engineering*, pages 328–338, 1987.

Dan Saffer. *Designing Gestural Interfaces*. O'Reilly Media Inc., 2008.

Kurt Schneider. LIDSs: A Light-Weight Approach to Experience Elicitation and Reuse. *Product Focused Software Process Improvement - Lecture Notes in Computer Sciences*, 1840: 407–424, 2000.

Kurt Schneider. *Experience and Knowledge Management in Software Engineering*. Springer Berlin Heidelberg, Berlin, Germany, 1st edition, 2009.

Till Schümmer and Stephan Lukosch. *Patterns for Computer-Mediated Interaction*. John Wiley & Sons, West Sussex, England, 2007.

Bill Scott and Theresa Neil. *Designing Web Interfaces*. O'Reilly Media Inc., Sebastopol, CA, USA, 2009.

Martin Sedlmayr. *Fraunhofer Series in Proaktive Assistenz zur kontextabhängigen und zielorientierten Unterstützung bei der Indikationsstellung und Anwendung von*. Doctoral thesis, RWTH Aachen University, 2008.

Eri Shimomukai, Sumire Nakamura, and Iba Takashi. Change Making Patterns: A Pattern Language for Fostering Social Entrepreneurship. In *Proceedings of the 19th Conference on Pattern Language of Programs (scheduled for 2014)*, 2014.

Ben Shneiderman and Catherine Plaisant. *Designing the User Interface: Strategies for Effective Human-Computer Interaction (4th Edition)*. Pearson Addison Wesley, 2004.

Jason Mcc Smith and David Stotts. Elemental Design Patterns: A Link Between Architecture and Object Semantics Elemental Design Patterns. In *Proceedings of the 27th Annual NASA Goddard Software Engineering Workshop (SEW-27'02)*, pages 183–199, Washington, DC, USA, 2002. Univ. of North Carolina at Chapel Hill, IEEE Computer Society.

Carolyn Snyder. *Paper prototyping*. Morgan Kaufmann, 2003.

Ian Sommerville, Tom Rodden, Pete Sawyer, and Richard Bentley. Sociologists can be surprisingly useful in interactive systems design. In *Proceedings of the conference on People and computers VII*, HCI'92, pages 342–354, New York, NY, USA, 1993. Cambridge University Press. ISBN 0-521-44591-4. URL http://dl.acm.org/citation.cfm?id=164592.164642.

Lucy Suchman. *Human-Machine Reconfigurations: Plans and Situated Actions (Learning in Doing: Social, Cognitive, and Computational Perspectives)*. Cambridge University Press, New York, New York, USA, 2nd edition, 2007.

Gerson Sunyé, Alain Le Guennec, and Jean-Marc Jézéquel. Design Pattern Application in UML. In *Proceedings of the 14th European Conference on Object-Oriented Programming (ECOOP '00)*, pages 44–62, London, UK, 2000. Springer.

Toufik Taibi. Formal Specification of Design Patterns - A Balanced Approach. *Journal of Object Technology*, 2(4):127–140, 2003.

Jenifer Tidwell. *Designing Interfaces*. O'Reilly Media, Sebastopol, CA, USA, 2nd edition, 2011.

Maryam Tohidi, William Buxton, Ronald Baecker, and Abigail Sellen. Getting the Right Design and the Design Right: Testing Many Is Better Than One. In *CHI '06 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1243–1252, 2006. doi: 10.1145/1124772.1124960.

Andres Toxboe. UI Patterns - User Interface Design Pattern Library. Available online: http://ui-patterns.com [Accessed on 26 September 2013].

Murray Turoff, Michael Chumer, Bartel Van de Walle, and Xiang Yao. The Design of a Dynamic Emergency Response Management Information System (DERMIS). *Journal of Information Technology Theory and Application (JITTA)*, 5(4):1–35, 2004.

UNITiD Interaction Designers. Android Patterns. Available online: http://www.androidpatterns.com [Accessed on 26 September 2013].

W.M.P. van der Aalst, W.M.P. ter Hofstede, B. Kiepuszewski, and A.P. Barros. Workflow Patterns. *Distributed and Parallel Databases*, 14(3):5–51, 2003.

Douglas K. van Duyne, James A. Landay, and Jason Hong. *The Design of Sites: Patterns for Creating Winning Websites*. Prentice Hall, 2nd edition, 2007.

Martijn van Welie. Welie.com - Patterns in Interaction Design. Available online: http://welie.com [Accessed on 26 September 2013].

Martijn van Welie and Hallvard Traetteberg. Interaction Patterns in User Interfaces. In *Proceedings of the Seventh Pattern Languages of Programs Conference PLoP 2000*, pages 13–16, 2000.

Stephen Viller and Ian Sommerville. Ethnographically informed analysis for software engineers. *International Journal of Human-Computer Studies*, 53(1):169–196, July 2000. doi: 10.1006/ijhc.2000.0370.

Larry E. Wood. Field Methods Casebook for Software Design. In *Field Methods Casebook for Software Design*, pages 35–56. John Wiley & Sons, Inc., New York, NY, USA, 1996.

Daniela Wurhofer and Marianna Obrist. A Quality Criteria Framework for Pattern Validation. *International Journal On Advances in Software*, 3(1):252–264, 2010.

Yahoo! Inc. The Yahoo! Design Pattern Library. Available online: http://developer.yahoo.com/ypatterns [Accessed on 26 September 2013].

Uwe Zdun. Systematic Pattern Selection Using Pattern Language Grammars and Design Space Analysis. *Software: Practice and Experience*, 37(9):983–1016, July 2007. doi: 10.1002/spe.799.

Alfred Zimmermann and René Reiners. Pattern Innovation for Architecture Diagnostics in Services Computing. In *PATTERNS 2012, The Fourth International Conferences on Pervasive Patterns and Applications*, pages 17–22. IARIA, XPS - Xpert Publishing Services, 2012.

Alfred Zimmermann, Fritz Laux, and René Reiners. A Pattern Language for Architecture Assessments of Service-oriented Enterprise Systems. In *PATTERNS 2011, Third International Conferences on Pervasive Patterns and Applications*, pages 7–12. XPS - Xpert Publishing Services, September 2011.

Index

Related Publications

The following publication list intends to show the development of ideas for the presented EPL concept throughout different cross-domain research activities.

Publications directly related to the topic of evolving design patterns:

René Reiners. Towards a Common Pattern Language for Ubicomp Application Design - A Classification Scheme for Ubiquitous Computing Environments -. In *PATTERNS 2010, The Second International Conferences on Pervasive Patterns and Applications*, pages 28–33. IARIA Conference, Think Mind(TM) Digital Library, November 2010.

René Reiners. An Extended Pattern Language Approach for UbiComp Application Design. In *Lecture Notes in Informatics - Informatiktage 2011*, page 4, Bonn, Germany, 2011a. GI - Gesellschaft der Informatik.

René Reiners. New Pattern Language Concepts for Designing UbiComp Applications Connecting to Cloud Services. In René Reiners and Alfred Zimmermann, editors, *Informatik 2011 : Informatik schafft Communities: Beiträge der 41. Jahrestagung der Gesellschaft für Informatik e.V. (GI), 4. - 7.10.2011 in Berlin*, pages 100–105, Bonn, Germany, September 2011b. GI - Gesellschaft für Informatik.

René Reiners, Irina Astrova, and Alfred Zimmermann. Introducing new Pattern Language Concepts and an Extended Pattern Structure for Ubiquitous Computing Application Design Support. In René Reiners and Alfred Zimmermann, editors, *PATTERNS 2011, Third International Conferences on Pervasive Patterns and Applications*, pages 61–66. XPS - Xpert Publishing Services, September 2011.

René Reiners. A Pattern Evolution Process - From Ideas to Patterns. In *Lecture Notes in Informatics Informatiktage 2012*, pages 115–118, Bonn, Germany, 2012. Gesellschaft für Informatik e.V., GI - Gesellschaft für Informatik.

René Reiners, Ragnhild Halvorsrud, Aslak Wegner Eide, and Daniela Pohl. An Approach to Evolutionary Design Pattern Engineering. In *Proceedings of the 19th international Conference on Pattern Languages of Programs PLoP 2012 (scheduled for 2014)*. ACM Digital Library, 2014b.

René Reiners, Michael Falkenthal, Dierk Jugel, and Alfred Zimmermann. Requirements for a Collaborative Formulation Process of Evolutionary Patterns. In *preparation to appear in the Proceedings of the 18th European Conference on Pattern Languages of Programs EuroPLoP '13 (scheduled for 2014)*. ACM Digital Library, 2014a.

Joint activities in the domain of emergency response and topics regarding collaboration and motivation in software engineering:

Christian Reinhard Prause, Marius Scholten, Andreas Zimmermann, René Reiners, and Markus Eisenhauer. Managing the Iterative Requirements Process in a Multi-National Project using an Issue Tracker. In *In 3rd International Conference on Global Software Engineering. IEEE*, pages 151–159, Bangalore, India, August 2008. IEEE Computer Society Press.

Christian Reinhard Prause, René Reiners, and Silviya Dencheva. Empirical Study of Tool Support in Highly Distributed Research Projects. In *Preceedings of the 2010 International Conference on Global Software Engineering (ICGSE)*, pages 23 – 32, Princeton / NJ, USA, August 2010b.

Christian R. Prause, René Reiners, Silviya Dencheva, and Andreas Zimmermann. Incentives for Motivating Scientists to Develop High-Quality Source Code. In *Preceedings of the Psychology of Programming Interest Group (PPIG) Workshop*, page 6 pp., Dundee, UK, 2010a.

Christian R. Prause and René Reiners. A Software Process Framework for PLE Development to Reduce Time and Costs of Widget Creation and Evolution. In *PLE Conference*, number 1988, page 5 pp., 2011.

Joint activities regarding patterns in the domain of service-oriented architectures:

Alfred Zimmermann, Fritz Laux, and René Reiners. A Pattern Language for Architecture Assessments of Service-oriented Enterprise Systems. In *PATTERNS 2011, Third International Conferences on Pervasive Patterns and Applications*, pages 7–12. XPS - Xpert Publishing Services, September 2011.

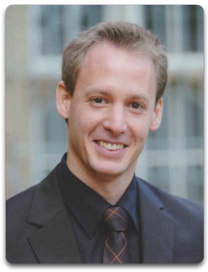
Michael Falkenthal, Dierk Jugel, René Reiners, Wilfried Reimann, and Michael Pretz. Maturity Assessments of Service-oriented Enterprise Architectures with Iterative Pattern Refinement. In *Informatik 2012 : Was bewegt uns in der/die Zukunft?: Beiträge der 42. Jahrestagung der Gesellschaft für Informatik e.V. (GI), 16. - 12.9.2012 in Braunschweig*, pages 1095–1101, Bonn, Germany, 2012. Gesellschaft für Informatik (GI).

Alfred Zimmermann and René Reiners. Pattern Innovation for Architecture Diagnostics in Services Computing. In *PATTERNS 2012, The Fourth International Conferences on Pervasive Patterns and Applications*, pages 17–22. IARIA, XPS - Xpert Publishing Services, 2012.

Curriculum Vitae

René Reiners

Fraunhofer FIT
Schloss Birlinghoven
53754 Sankt Augustin, Germany
Phone: +49 (0) 2241 / 14-2615
eMail: rene.reiners@{fit.fraunhofer.de, rwth-aachen.de}



Personal Data:

Date of Birth:	May 27, 1980
Birth Place:	Mönchengladbach-Rheydt, Germany
Nationality:	German

Education:

1986 - 1990:	Catholic Primary School Ratheim 41836 Hückelhoven, Germany
1990 - 1999:	Cusanus Gymnasium 41812 Erkelenz, Germany Degree: "Abitur"

Civilian Services:

1999 - 2000:	Caritas Association Aachen e.V., Germany Caritas Care Station Erkelenz 41812 Erkelenz, Germany

Academic Education:

2000 - 2006	RWTH Aachen University 52062 Aachen, Germany Graduation in Computer Science Degree: "Diplom-Informatiker"
Studies Focus:	Human-Computer-Interaction Design Patterns Object Oriented Software Construction Software-Quality Assurance and Project Management Database Management Systems

Professional Experience:

2006 - 2007:	Organization and Development at REWE – Information Systems (RIS) GmbH Humboldtstr. 144 51449 Cologne, Germany
since 2007:	Research Associate and Project Manager at the Fraunhofer Institute for Applied Information Technology FIT Schloss Birlinghoven 53754 Sankt Augustin, Germany

Language Skills:

German (Native)
English (C1 - Effective Operational Proficiency)
French (B2 - Vantage)

Professional Skills:

OMG - Certified UML Professional
IREB - Certified Professional for
Requirements Engineering
Certified Project Management Professional, PMP

Technical Skills:

OO Programming:	Java, Objective-C
Web Programming:	PHP 5, JSP, HTML, CSS, JavaScript, Apache Webserver, Apache Tomcat
DBMS:	IBM DB2, MySQL, Microsoft Access
Frameworks:	Hibernate, Struts, Joomla!
IDEs:	Eclipse, XCode, IntelliJ
Tools:	Atlassian JIRA, Atlassian Confluence, GForge, MediaWiki, CVS, Subversion, MS Project, MS Office Suite, Open Office Suite
Operating Systems:	Windows, Mac OS X, Linux

Hobbies:

Reading, Languages, Badminton, Biking, Jogging

Typeset December 12, 2013