Requirements for a Collaborative Formulation Process of Evolutionary Patterns

René Reiners[1], Michael Falkenthal[2], Dierk Jugel[2], Alfred Zimmermann[2]

[1]Fraunhofer FIT, Germany,
rene.reiners@fit.fraunhofer.de

[2]University of Applied Sciences Reutlingen, Germany,
firstname.lastname@reutlingen-university.de

BibTeX:

```
@inproceedings{Reiners13,
  author    = {Ren\’{e} Reiners and Michael Falkenthal and Dierk Jugel and
               Alfred Zimmermann},
  title     = {Requirements for a Collaborative Formulation Process
               of Evolutionary Patterns},
  booktitle = {Proceedings of the 18th European Conference on Pattern Languages
               of Programs (EuroPlop)},
  year      = {2013},
  pages     = {1–10}
}
```

Requirements for a Collaborative Formulation Process of Evolutionary Patterns

René Reiners, Fraunhofer FIT
Michael Falkenthal, Reutlingen University
Dierk Jugel, Reutlingen University
Alfred Zimmermann, Reutlingen University

Within distributed research projects, it is necessary to align many kinds of engineering efforts across different work packages within the considered domain. Every project member has his own visions, expectations and experience that must be considered throughout the whole project duration. This work addresses challenges encountered in distributed research projects in which the research consortia consist of partners from research and industry with multidisciplinary backgrounds. Different working practices and documentation as well as specification material is produced by every partner. We aim at giving every project member the chance to contribute knowledge in an easy and light-weight manner right from the start of the project. Our approach applies the idea of using design patterns as interdisciplinary communication medium, especially for describing and explaining the current application domain and its characteristics.
This work presents considerations from two workshops with participants from two different projects that work in the domain of emergency response and enterprise & service-oriented architectures. The aim of the workshops was to analyze current pattern identification and pattern language structuring processes concerning their suitability to engineering practices within distributed research projects. Although the idea of using patterns as interdisciplinary communication medium was considered as useful, challenges that were faced with current pattern derivation processes were extracted. In a second step, the workshop participants agreed on three sets of requirements that a collaborative pattern formulation process should fulfill. The third set of requirements is especially suited to the domain of service-oriented architectures.

Categories and Subject Descriptors: H.5.3 [**Information Interfaces and Presentation**]: Group and Organization Interfaces—*Computer-supported cooperative work*; K.4.3. [**Organizational Impacts**] Computer-supported collaborative work

General Terms: Human Factors

Additional Key Words and Phrases: Design Patterns, Pattern Languages, Pattern Formulation, Collaboration

1. INTRODUCTION

The considerations of this work concentrate on aspects of knowledge management and exchange within distributed research projects. The working set of our efforts is based on current projects, i.e. on the one hand side the EU funded, international project BRIDGE[1] running since April 2011 for four years, and on the other hand the SOA Innovation Lab[2] that was founded in 2012 as a German, national research consortium in the domain of service-oriented architectures.

The BRIDGE project seeks for solutions to support crisis management and interoperability for technical and social aspects in large-scale emergency management and therewith increasing the public safety. In the problem scope, multiple First Responder organizations in Europe need to collaborate and coordinate common activities.

[1] http://bridgeproject.eu
[2] http://soa-lab.de

Possible large-scale scenarios like technological disasters or natural catastrophes and threats like terrorist attacks frame the research activities. 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 disaster simulations.

The SOA Innovation Lab allows the connection between applied practices in industry and more theory-related academic foundations. Together with 15 members from industry in Germany and Switzerland and additional partners from academia, the challenges of good software architecture designs in the context of applying new technologies and paradigms such as Cloud Computing and proven evaluation methods are addressed in order to provide them as a key competence of the consortium.

The considered project consortia usually consist of around 15 project partners from academia and industry. This way, the number of total participants quickly spans between 50 and 100 people. The high number of participants and individual partner's specializations including experience, technology and processes lead to high communication overheads in situations in which changes on the pursued strategy, formulated requirements or derived concepts and realizations need to be made. Large amounts of documentation must be available in manageable sizes in order to be effective. In current situations but also in retrospective, reasons for design decisions must be understandable and reproducible. In large scale distributed research projects, the proposed ideas for system and concept designs become more difficult to apply over time. Collaboration and coordination of knowledge integration into the whole project scope becomes difficult to handle. Especially in research projects, knowledge transfer and reintegration are crucial aspects regarding synergies and newly combined findings. In order to manage efforts and tasks between the different stakeholders of the projects, work packages are created that are further subdivided into tasks. The work packages follow dedicated purposes like *domain analysis*, setting up the *system's architecture*, *conception* and *implementation* as well as *validation* and *exploitation* of the results.

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 derived 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 fed back to the project-wide knowledge repository. This also affects implementation plans since it is possible that validations reveal that agreed concepts need to be changed or even be replaced. 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 findings 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 need to be accumulated in the general project's knowledge pool from which the appropriate work packages extract required knowledge in a format that is suited to their demands.

In previous work we discussed the difficulties regarding communication overhead, knowledge generation, awareness and exchange between the different work packages (cf. [Reiners et al. 2012]). The problem was also seen in a survey conducted by [Prause et al. 2010] who present results from a survey on research projects within the Sixth EU Framework Programme[3] in which 152 representatives from 73 different projects participated. Extrapolation in total regarded 741 projects out of 1167 as relevant for the study. The evaluation reveals that communication within the projects takes place a lot, predominantly via emails, phone calls and chats. Still, the documentation that is exchanged is attached to the communication channels without preparation for the recipient. Interested parties need to actively transform the formats to their needs.

[3] http://ec.europa.eu/research/fp6/pdf/fp6-in-brief_en.pdf

2. APPROACH: DESIGN PATTERNS AS MICRO-DOCUMENTATION

Our approach to improve the knowledge generation and exchange within distributed research projects applies the idea of using patterns as interdisciplinary communication medium for describing and explaining the current application domain, its characteristics and derived application concepts. The approach aims at improving the mutual understanding and communication between work packages especially concerned with domain analysis, design and validation. Additionally, other project members also benefit from the descriptions such that, even if their work is very technology-oriented, they have an easier access to the gathered information and derived concepts. This way, the understanding of *why* a specific concept has to be implemented in the given way can be conveyed. By applying the design pattern concept, an additional aim is to involve more stakeholders in the knowledge engineering process besides the individual work packages' experts.

Many existing pattern languages aim at keeping the formulations of the single patterns very readable and easy to understand. This way, patterns can serve as a *lingua franca* between stakeholders and interested parties new to the domain. [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. The aim is to improve the exchange between different disciplines with a common vocabulary that gives project members more expressiveness concerning the problem domain.

Traditionally, design patterns are formulated *after* gathering a lot of experience by experts over time and providing them for upcoming challenges. The approach we follow incorporates work steps *during the project research work* concerning the formulation of existing knowledge, rules and processes as well as *new approaches* that are currently under development as patterns. Concepts for managing the formulation and dissemination process as well as the evolution of a design pattern together with the management of the whole pattern repository are developed. As discussed in previous work, our approach applies the concept of pattern languages concept as *evolutionary micro-documentation* throughout the engineering and knowledge management process (cf. [Reiners et al. 2011; Reiners 2012; Zimmermann and Reiners 2012; Reiners et al. 2012]). Instead of formulating approved solutions resulting from long-term experience, every finding and idea is formulated as an, initially incomplete, ”proto pattern” that matures in parallel to the project lifetime. The concept aims at enabling project participants to quickly formulate, review, comment and refine existing pieces of knowledge and provide evidence for or against the described concept.

3. ENCOUNTERED PROBLEMS

Joint activities between Fraunhofer FIT[4] and the research group on Enterprise and Software Architectures at Reutlingen Univeristy[5] (ARL) revealed similar interests in formulating project and research knowledge as patterns. However, both groups experienced similar challenges for the formulation process and collaboration regarding maintenance and extending existing repositories. Generated knowledge originating from research and industry projects is regularly collected as patterns in the reduced canonical form within pattern catalogs as described by [Zimmermann et al. 2011] in the scope of assessing the maturity of software architectures.

Common considerations in both research groups and within the two different project backgrounds resulted in the demand for the conception of a collaborative approach to *iteratively* formulate patterns into a developing pattern library structure. First considerations were already formulated in a high-level description of a collaborative formulation process of evolutionary patterns in [Reiners et al. 2012] that acts as the starting point for realizations within the two project contexts. This work reflects the problem and requirements analyses that were performed in two expert workshops that compared existing pattern mining approaches for published pattern languages and formulated needs of the envisioned process. More traditional printed publications and the applied pattern mining approaches were analyzed together with online repositories that allow for more interaction between pattern

[4] http://fit.fraunhofer.de/en.html

[5] http://www.reutlingen-university.de/forschung/forschungszentren-gruppen/serviceorientierte-software-architekturen.html

authors and readers. The approaches were compared with the needs for iterative and collaborative pattern formulation in the scope of research projects. The overall result states that existing approaches only partially support the demands for this special kind of project and knowledge acquisition and are expressed as a collection of problems presented in the following section. From the problems with existing approaches together with the needs of the individual projects' domains, requirements are then formulated in Section 5 that lay the foundation for the conception and realization of a collaborative formulation process of evolutionary patterns that is addressed in future work in Section 6.

4. SHORTCOMINGS OF TRADITIONAL PATTERN MINING PROCESSES

Representatives from both research consortia decided to exchange on problems, requirements and findings in the particular domains and merge commonalities. The common efforts are reflected by two joint workshops on problems of current pattern formulation and organization processes with regard to using them as micro-documentation within ongoing research activities. From the problems, mandatory results for a collaborative and iterative pattern formulation process were derived as a basis for upcoming realizations of a pattern-supported engineering process accompanied by individual pattern portals.

We arranged two joint expert workshops that were held in September 2011 and June 2012. The first workshop was each held with eight participants from the BRIDGE project that are involved in 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 as described in Section 4.1 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 provided 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. The following section summarizes the regarded pattern collections and languages in written and online publications before Section 4.2 summarizes the results of the comparison and the formulated problems of existing approaches with regard to envisioned approach in the scope of research projects.

4.1 Regarded Pattern Collections and Languages

The comparison took into account prominent examples of pattern collections and languages that were also formulated in domains different from software engineering. As the common root of all approaches, the work of Christopher Alexander was analyzed in which he formulated suggestions on environmental arrangements for building architecture [Alexander 1977]. His pattern language starts with very general concepts about structuring city sites and further reveals more details about districts, quarters up to single kinds of shops and houses until very detailed concepts for single rooms and environmental aspects. The whole pattern language is written in prose and avoids domain jargon such that every interested party is able to understand the concepts without the need for expert knowledge.

[Gamma et al. 1994] reused the idea again and formulated design patterns for software engineering. Their famous work is often reused until today in practice and teaching. Since the formulated patterns deal with the details of software construction, the thought of general understandability is given up.

Later work by [Tidwell 2011; Coplien and Harrison 2005; Manns and Rising 2005] and [Borchers 2001] reintroduce readability for large readerships. They present pattern languages for interface design, organizational structures and processes as well as interactive exhibit design, respectively. More examples are found in the domain of education (cf. [Bergin 2012]) or collaborative system design (cf. [Schümmer and Lukosch 2007]).

Especially in the scope of the regarded projects, patterns for emergency response, in particular working practices of firefighters, were formulated by [Denef 2011]. For system and interface design in safety-critical environments, the approaches of [Hussey 1999; Connelly et al. 2001; Carver and Turoff 2007; Grill and Blauhut 2008; Nilsson 2009; Acuna et al. 2010] are also described as patterns. The mentioned pattern collections are candidates for the pattern repository envisioned for the BRIDGE project.

The ARL research group at Reutlingen University aims at formulating, managing and maintaining gathered knowledge as patterns and making it available within the SOA Innovation Lab consortium but also for teaching purposes within their group. Approaches regarding the assessment of the maturity of software architectures are formulated and published as a catalog in [Zimmermann et al. 2010] and as a pattern language in [Falkenthal et al. 2012; Zimmermann and Reiners 2012]. Another major topic of the ARL focuses the domain of cloud computing patterns like presented in [Fehling et al. 2012]. Finally, the area of enterprise architecture management patterns such as presented in [Buckl et al. 2012] are investigated. T eaching efforts and internal work groups continuously research on knowledge in the SOA domain and formulate them as, still independent, pattern catalogs.

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 that can partially be handled by user-comments on the patterns and moderators that maintain the library. Most of the online available pattern collections deal with commonly applicable user interface 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. [van Welie 2008; Toxboe 2012; Yahoo! Inc. 2012; PatternTap LLC 2012]). 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 their position within the pattern libraries. Still, it is possible to navigate towards specific design problems in a straight forward manner. The aim of these platforms is to provide support on specific problems and less on providing the large situational picture of the domain.

The *Quince Pattern Library* and *Patternry* represent exceptions to the public community process since the free versions provide publicly available patterns but try to convince the user to register for a chargeable extended license that provides collaboration spaces restricted to project teams (cf. [Infragistics 2012; Lammi et al. 2012]).

4.2 Identified Problems With Existing Approaches

The result of the assessment of existing pattern mining processes is summarized in Table I that compares printed and web-based publications of patterns. For patterns that were published in a printed format, the formulation happened over after a long period of research that was lead by the authors. With the growing importance of the world-wide Web in the late 90s and together with collaborative mechanisms that were developed on top of the Web infrastructure, pattern portals were realized. The comparison reflects the interdisciplinary aspect of the pattern approach since it is not exclusively applied in a single domain. During the workshops, the current approaches were considered as not fully supporting the needs for the envisioned collaborative formulation process for evolutionary patterns in the scope of research projects. The comparison shows that current pattern mining and formulation process still face a number of challenges that need to be addressed in the scope of the envisioned pattern-engineering approach. The different symbols $\ominus\ominus$, $\ominus$, $\odot$, $\oplus$ and $\oplus\oplus$ reflect the average perception of features that were examined within the analyzed collections and languages. $\ominus\ominus$ stands for no support of the considered feature, $\ominus$ 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. $\oplus$ and $\oplus\oplus$ are attributed for features that are strongly supported or in an outstanding way.

	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 ⇒ Focus on knowledge sharing
Structure	Clusters by purpose, behavior, structure, category, topic, development step	Hierarchies and dependencies on spatial or temporal aspects, implementation detail, task order
Collaboration & Influence	⊖⊖	⊙
Extensibility	⊖⊖	⊕
Reusability	⊙	⊙
Transparency	⊖⊖	⊙
Bad Practices	⊖⊖	⊖
Recommendation	⊙	⊕
Guidance	⊕	⊕
Mining Process	(Internal) assessment, mining workshops, partially shared on conferences ⇒ Based on experience.	Online discussion based on commenting, rating and voting. Community-based proposals.

Table I. : A feature comparison of patterns publications in printed formats and online portals.

From the assessment of currently available approaches, a list of extracted problems was created during the workshops that is shown in Tables II and III. The problem list represents the starting point for the requirements elicitation described in the next Section.

P1– Tedious Pattern Generation: Currently, patterns are gathered over time by small groups of experts that have completely verified their findings before publishing them as 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 patterns. For teaching and production purposes, this is undoubtedly a good approach since the patterns are intended to provide knowledge that is ready to use. For the pursued collaborative formulation process however, workshop participants found it useful to document the whole way of idea finding, refining and validation of a pattern.	*P2– Closed Author Groups:* The traditional pattern mining and structuring process in collections or pattern 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 organization structure. [Reiners 2012] call this the ”Pattern Guru Approach” in an intentionally provoking manner since common collection and formulation of knowledge is rarely performed.
P3– Lacking Transparency of Derivation: In written publications, patterns undergo internal assessment processes within the authors' working environments. In any case, the derived patterns represented a reviewed and valuable source of knowledge shared by experts. The derivation of a formulated pattern was considered as relevant for the readers of the patterns. Ratings for patterns must be comprehensible, otherwise, they resemble more a rule of thumb instead of a validation. Online portals allow readers to track the process of ratings that are supported by users. Inserted patterns are commented and discussed collaboratively. Still, the derivation of the pattern itself is not shown but only a mature state that can be accepted or rejected.	*P4– Lacking Reuse of Existing Knowledge:* Current pattern languages describe evaluated solutions to problems in specific domains. Though, especially for conceptual patterns at a higher abstraction levels, 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. Lots of relevant knowledge is bound in publications, articles, internal studies, project reports or products. This affects recommending successful approaches and the availability of knowledge about bad designs.

Table II. : Summary of extracted problems from the feature comparison during the workshops - Part 1 of 2.

P5– Lacking Recommendation and Guidance: 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 clusters of hierarchy levels for navigating through the pattern language but some lack from visualizations of the entire structure.

P6– Lacking Knowledge of Bad Practices: Published patterns, often reveal working concepts. Only initial studies about a certain problem domain concretely outline deficits in order to justify and motivate intended research. During the workshops, participants underlined the importance of non-trivial bad practices that are revealed unexpectedly during experiments and realizations.

P7– Lacking Influence on Pattern Formulation: Currently, there are only basic possibilities to feed experience back into existing pattern languages. The discussion on existing patterns is still hard to trigger and keep alive. Published patterns remain relatively static. 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 that the publications are made via books, the reintegration has to be postponed to the next edition. 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 practicioners who applied the pattern. New ideas and concepts cannot be included into the existing structure and thus influence a pattern's rating, refinement and validity.

P8– Extensibility of Pattern Language: Currently, patterns undergo a discussion process either together with domain experts directly or via workshops that support the understandability and applicability of a pattern. At some point in time, authors have to stop and present the current state of the derived patterns. Without claiming that their patterns are complete, they need to fix the derivation process for the 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. Again, web-based approaches were considered as being capable to allow future discussion and drive the derivation process further. Validated patterns need to be reassessed over time in order to check for possible updates or extensions.

Table III. : Summary of extracted problems from the feature comparison during the workshops - part 2 of 2.

5. IDENTIFIED SETS OF REQUIREMENTS

After the comparison of relevant features, the problems that were seen with current approaches were extracted that reflect the concerns and opinions that were stated during workshop discussions. From this basis of concerns, three sets of requirements were formulated.The basic requirements that need to be fulfilled are listed in Table IV that 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 patterns from external collections. The development state of a pattern should be reflected based on its applicability and validity.

R1– Availability of Knowledge: The assembly and refinement of knowledge should be encapsulated within patterns and always be available in any development state.

R2– Early Participation of Stakeholders: The approach should include all stakeholders during the whole project lifetime in the collaboration and contribution process.

R3– Quick Contributions: Easy ways to access the pattern repository 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.

R4– Liveliness and Extensibility: Easy ways of formulating newly discovered results and updating for existing patterns are needed. Feedback on propositions should be given directly and be available for every project member, independently on time and location.

R5– Reuse of Knowledge: Mechanisms are needed to insert and adapt already proven design problems in the pattern repository. Adaptation may be needed since the original pattern was eventually suited to special needs of a specific domain.

R6– Pattern Maturity: A flexible rule system needs to defined that determines 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.

Table IV. : Basic requirements for a collaborative pattern formulation portal.

The second set of requirements that are listed in Table V focuses on ways to better support the community-based pattern formulation process. Aspects on visualization as well as the development of the structure of a design pattern repository over time became more important. Domain knowledge especially including results from validated approaches should be influenced in a public way. Project members need to be encouraged to contribute their knowledge at all states of development and validation since the aim is to reflect the individual group's work.

R7– Pattern Validity: Mechanism to provide the inclusion of evidence in favor of or against a formulated pattern should be integrated into the evolutionary pattern derivation process. Votes for or against successful applications of the patterns should be collected from practitioners.	*R8– Decision Support:* Starting from an entry point within the pattern language structure, ranking-based guidance through a sequence of patterns should be provided via showing pattern combinations that were often chosen by the users of the pattern repository.
R9– Structural Development: A flexible hierarchy concept must be developed that is iteratively developed and refined over time, similar to the single patterns. Structural changes, introduction of new hierarchy levels as well as removal of existing hierarchies without affecting the pattern formulations should be realized.	*R10– Anti-Patterns:* Non-trivial, surprisingly failing approaches should be documented as anti-patterns within the pattern language. Existing patterns may develop towards anti-patterns and vice versa.
R11– Visualization and Orientation: Smart visualization methods are needed to support the users' browsing and contribution activities regarding the current state of a pattern, the pattern repository structure, a pattern's position within the structure and relations between the patterns.	*R12– Transparency of Process:* The evolutionary design pattern repository must provide means of reflecting the community's activities, new contributions as patterns, examples, solutions or evidence as well as newly added structural elements. Neglected patterns need to be reconsidered by the community again.

Table V. : Overview of advanced requirements for a collaborative pattern formulation process.

In addition to the two sets of basic and advanced requirements, special needs in the SOA domain were addressed by the ARL group. Needs for management, control and overview of the existing patterns play an important role since the group already formulated many mature patterns that are applied in practice.

R13– Semantic Enrichment of Patterns: Since a plenty of several pattern repositories existing in the web lead to a huge amount of redundancy of patterns the idea is to add semantic meta information to patterns. Enriching patterns could lead to easier connections between different pattern repositories or even to completely new repositories	*R14– Pattern KPIs:* In order to provide a general overview of the pattern maturity it was stated to be important to introduce pattern KPIs. Defining KPIs for the completeness and maturity of patterns could lead to more manageable pattern catalogs once a catalog is populated with many patterns.
R15– Management Cockpit: Introducing general pattern management information in the form of pattern and pattern repository KPIs open out into a view needed to see the defined KPIs,	*R16– Authorship and Reputation:* Authorship, i.e., the connection between individual ideas and contributions should be taken into account such that intellectual property is preserved.
R17– Management of Repository Contents: A defined process of managing the content of the pattern repository must ensure the repository's actuality.	*R18– Structural Elements:* Additional relations that express *AND* as well as *XOR* semantics should further structure the repository as soon as the number of patterns grows.

Table VI. : Overview of requirements especially formulated for the ARL group at Reutlingen University.

6. CONCLUSION AND FUTURE WORK

The aim of the approach presented in this work 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 are needed in order to combine the concept with the special characteristics of research projects. Documentation is generated during the project work and extends or refined existing results. External influences from other projects and related research also need to be integrated.

An exchange of gathered experience between members of the EU-funded BRIDGE project and the research group on Enterprise and Software Architectures at Reutlingen University revealed that similar demands and problems concerning knowledge formulation and exchange between consortium members occurred. Both groups consider patterns as a possible medium for formulating knowledge. Thus in two cooperative workshops, requirements were derived that result from two different practical projects from different domains, i.e., service-oriented architectures and emergency response. Though different backgrounds, the general demands and problems that were collected during the workshops were similar and could be integrated into one common approach.

Two workshops with members from the domain of emergency response and service-oriented architectures analyzed currently available pattern mining and structuring approaches. The extracted problems of existing approaches were fed into the second part of the workshop in which three sets of requirements for a collaborative pattern formulation process were formulated.

The basic set of necessary requirements was validated and realized in two prototypical implementations of the BRIDGE Design Pattern Library[6] and the ARL Pattern Repository prototype[7]. The repositories currently hold 26 and 43 patterns respectively that were gathered during the BRIDGE project work steps and catalogs derived from an evaluation project of service-oriented capabilities of leading ERP vendors at Reutlingen University and the SOA Innovation Lab.

A next design iteration for both prototypes will incorporate the advanced sets of requirements that are formulated in Section 5. From the requirements, a generic process for an evolutionary pattern library will be formulated and concentrate on the realization of pattern evolution states, derive a light-weight role model for access control and track changes. Additionally, concrete measures for a pattern's maturity and validity will be defined. The ARL prototype will additionally address the specialized set of requirements that address characteristics of the projects the ARL group participates in. The upcoming iteration of the envisioned approach is a mixture of ongoing research work and a collection of practical experience with two real platform. From the experience and lessons learned, the generic conception will be adapted.

7. ACKNOWLEDGEMENTS

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement n^{o}261817, the BRIDGE project (www.bridgeproject.eu).equipment.

Special thanks go to our shepherds Uwe van Heesch and Elisaveta Gurova for the provision of valuable advice on iteratively refining the paper.

REFERENCES

ACUNA, P., DIAZ, P., AND AEDO, I. 2010. Development of a Design Patterns Catalog for Web-based Emergency Management Systems. In *Proceedings of the 7th International ISCRAM Conference.* ISCRAM.

ALEXANDER, C. 1977. *A Pattern Language: Towns, Buildings, Construction.* Oxford University Press, New York, NY, USA.

BERGIN, J. 2012. *The Pedagogical Patterns Project.* CreateSpace Independent Publishing Platform.

BORCHERS, J. 2001. *A Pattern Approach to Interaction Design* 1st Ed. John Wiley & Sons, West Sussex, England.

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

[7] http://www.arlab.org (to be published in May 2013)

BUCKL, S., MATTHES, F., MONAHOV, I., ROTH, S., SCHULZ, C., AND SCHWEDA, C. M. 2012. Enterprise architecture management patterns for company-wide access views on business objects. In *Proceedings of the 16th European Conference on Pattern Languages of Programs*. EuroPLoP '11. ACM, New York, NY, USA, 11:1–11:12.

CARVER, L. AND TUROFF, M. 2007. Human-computer interaction: the human and computer as a team in emergency management information systems. *Communications of the ACM 50,* 3, 33–38.

CONNELLY, S., BURMEISTER, J., MACDONALD, A., HUSSEY, A., AND DREW. 2001. Extending and evaluating a pattern language for safety-critical user interfaces. *Conferences in Reseach and Practice in Information Technology 3*.

COPLIEN, J. O. AND HARRISON, N. B. 2005. *Organizational Patterns of Agile Software Development*. Pearson Prentice Hall, Hamilton, NY, USA.

DENEF, S. 2011. A Pattern Language of Firefighting Frontline Practice to Inform the Design of Ubiquitous Computing. Ph.D. thesis, TU Delft, Netherlands, RWTH Aachen Unversity, Fraunhofer FIT.

ERICKSON, T. 2000. 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. ACM, New York, NY, USA, 357–368.

FALKENTHAL, M., JUGEL, D., REINERS, R., REIMANN, W., AND PRETZ, M. 2012. 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*. Gesellschaft für Informatik (GI), Bonn, Germany, 1095–1101.

FEHLING, C., LEYMANN, F., RÜTSCHLIN, J., AND SCHUMM, D. 2012. Pattern-Based Development and Management of Cloud Applications. *Future Internet 4,* 4, 110–141.

GAMMA, E., HELM, R., JOHNSON, R. E., AND VLISSIDES, J. 1994. *Design Patterns. Elements of Reusable Object-Oriented Software* 1 Ed. Addison-Wesley Longman, Amsterdam.

GRILL, T. AND BLAUHUT, M. 2008. Design Patterns Applied in a User Interface Design (UID) Process for Safety Critical Environments (SCEs). In *HCI and Usability for Education and Work*, A. Holzinger, Ed. Lecture Notes in Computer Science Series, vol. 5298. Springer Berlin / Heidelberg, 459–474.

HUSSEY, A. 1999. Patterns for safety and usability in human-computer interfaces. Tech. rep.

INFRAGISTICS. 2012. The Quince Pattern Library. Available online: http://quince.infragistics.com [Accessed on 18 February 2013].

LAMMI, J., VARJOKALLIO, M., AND HOCKSELL, J. 2012. Patternry. Available online: http://patternry.com [Accessed on 18 February 2013].

MANNS, M. L. AND RISING, L. 2005. *Fearless Change: Patterns for Introducing New Ideas: Introducing Patterns into Organizations* 2005 Ed. Addison-Wesley, Boston, MA, USA.

NILSSON, E. G. 2009. Design patterns for user interface for mobile applications. *Advances in Engineering Software 40,* 12, 1318–1328.

PATTERNTAP LLC. 2012. Pattern Tap. Available online: http://patterntap.com [Accessed on 18 February 2013].

PRAUSE, C. R., REINERS, R., AND DENCHEVA, S. 2010. Empirical Study of Tool Support in Highly Distributed Research Projects. In *Preceedings of the 2010 International Conference on Global Software Engineering (ICGSE)*. Princeton / NJ, USA, 23 – 32.

REINERS, R. 2012. A Pattern Evolution Process - From Ideas to Patterns. In *Lecture Notes in Informatics Informatiktage 2012*. Gesellschaft für Informatik e.V., GI - Gesellschaft für Informatik, Bonn, Germany, 115–118.

REINERS, R., ASTROVA, I., AND ZIMMERMANN, A. 2011. Introducing new Pattern Language Concepts and an Extended Pattern Structure for Ubiquitous Computing Application Design Support. In *PATTERNS 2011, Third International Conferences on Pervasive Patterns and Applications*, F. Laux, R. Reiners, and A. Zimmermann, Eds. Number c. XPS - Xpert Publishing Services, 61–66.

REINERS, R., HALVORSRUD, R., WEGNER EIDE, A., AND POHL, D. 2012. Design Pattern Engineering: An Evolutionary Approach Applied to the Emergency Response Domain. In *To appear in the 19th Conference on Pattern Languages of Programs*. Tucson, Arizona, USA.

SCHÜMMER, T. AND LUKOSCH, S. 2007. *Patterns for Computer-Mediated Interaction*. John Wiley & Sons, West Sussex, England.

TIDWELL, J. 2011. *Designing Interfaces* 2nd Ed. O'Reilly Media, Sebastopol, CA, USA.

TOXBOE, A. 2012. UI Patterns - User Interface Design Pattern Library. Available online: http://ui-patterns.com [Accessed on 18 February 2013].

VAN WELIE, M. 2008. Welie.com - Patterns in Interaction Design. Available online: http://welie.com [Accessed on 18 February 2013].

YAHOO! INC. 2012. The Yahoo! Design Pattern Library. Available online: http://developer.yahoo.com/ypatterns [Accessed on 18 Feb 2013].

ZIMMERMANN, A., AMMANN, E., AND LAUX, F. 2010. Pattern Catalog for Capability Diagnostics and Improvement of Service-oriented Enterprise Architectures. In *IARIA Proceedings of the PATTERNS 2010 - The Second International Conferences on Pervasive Patterns and Applications*. Lisbon, Portugal, 13–19.

ZIMMERMANN, A., BUCKOW, H., GROSS, H.-J., NANDICO, O. F., PILLER, G., AND PROTT, K. 2011. Capability Diagnostics of Enterprise Service Architectures using a dedicated Software Architecture Reference Model. In *IEEE SCC'11*. 592–599.

ZIMMERMANN, A. AND REINERS, R. 2012. Pattern Innovation for Architecture Diagnostics in Services Computing. In *PATTERNS 2012, The Fourth International Conferences on Pervasive Patterns and Applications*. IARIA, XPS - Xpert Publishing Services, 17–22.