Sharing Requirements Engineering Experience Using Patterns

Lars Hagge and Kathrin Lappe, *Deutsches Elektronen-Synchrotron*

Many practitioners—especially those in small and medium enterprises—work in organizations that don't yet implement dedicated requirements engineering methods. Furthermore, these organizations often lack the appropriate knowledge and tools for implementing RE, so practitioners are confronted with

- Working with a customer organization that's not used to the interplay of IT systems and business processes and doesn't have established engineering processes
- Working with limited RE resources and having to convince customers and their users of RE's necessity
- Adopting RE at the project level because the project leader generally has little—if any—influence on the higher-level customer organization

For this target group of practitioners, we propose patterns as a format for RE knowledge transfer, which can provide guidance by offering easy access to proven methods and tools. As an example, we provide four patterns for basic RE activities.

Capturing requirements engineering experience as patterns enables knowledge transfer for practical use. Four introductory patterns for basic RE activities offer easy access to proven methods and tools, and a proposed analysis method explains how to derive patterns from observations.

Patterns for RE knowledge transfer

Patterns are an established and well-known format for capturing engineering knowledge. The idea originated in civil engineering in a series of books by Christopher Alexander, who observed that well-accepted buildings—which he called "alive"—have common characteristic structures.[1,2] He then developed a set of rules for architects on how to construct such buildings. The "Gang of Four" extended the pattern format to software design,[3] and others have further spread it to domains such as software architecture[4] and business processes[5] (see http://hillside.net/patterns for more information on patterns).

Today, practitioners use patterns to describe reference solutions to engineering problems and

0740-7459/05/$20.00 © 2005 IEEE

as guidelines for engineering procedures. Because our objective is to transfer RE knowledge at the practical-experience level, we focus on the latter use. Understanding "engineering" as the art of applying scientific and methodological knowledge to practical problems, patterns for RE activities should primarily contain guidelines for engineers—they may contain reference solutions as additional examples where necessary.

At first glance, patterns, best practices,[6] and processes all seem adequate for our objective. However, we consider patterns to be less formal than processes and more focused than best practices. Because processes are defined, agreed upon, and established within an organization at the management level, they're out of reach for project leaders. Best practices, on the other hand, can grow into large collections of tools, templates, guidelines, or anything else considered useful for a given task, and thus overwhelm the inexperienced.

To avoid such problems, we require that patterns are

- *Relevant,* addressing important RE decisions in projects rather than providing basic textbook information
- *Applicable,* containing guidelines that the project leader can implement without requiring support from higher-level organizations
- *Reliable,* having been successfully used under different conditions rather than originating from academic reasoning alone

Four patterns for basic RE activities

We present four patterns for basic RE activities as an introductory example. The patterns we describe can be seen as a small-scale RE starter kit: they offer guidance for organizing the specification procedure and for eliciting, specifying, and verifying requirements. We observed these patterns in a plant construction project and in a set of comparable projects introducing COTS-based systems, demonstrating that we can successfully transfer RE methods and tools beyond software engineering. The projects were carried out at the Deutsches Elektronen-Synchrotron (DESY) in Hamburg, Germany (www.desy.de), a research center for accelerator and particle physics and for scientific applications of synchrotron radiation.

Project #1 involved obtaining approval for a large new accelerator, which required creating an initial design of the planned buildings and installations (see http://xfel.desy.de). A core team of about 20 members prepared the design, with input from up to 100 experts from DESY and external engineering offices. The project management organized the core team and experts into working groups and introduced RE to the core team to help coordinate these groups.

Project #2 comprised a set of comparable projects introducing COTS-based information systems at DESY, such as product data management or facility management.[7] The projects were realized by internal IT and domain experts from different user groups, external developers from the COTS-system vendors, and general external consultants. The project teams typically consisted of 10 to 20 people, with only the external people (two to five) working full time on the projects.

To introduce the four patterns, we follow the formats of established pattern collections. We've kept the descriptions short; more elaborate versions appear elsewhere.[8,9]

RE Pattern 1: Organize the Specification along the Project Structure

This pattern recommends a way to organize the specification procedure.

Objective. Create a mutually agreed-upon requirements specification in an environment of distributed stakeholder groups and project teams.

Context. Project leaders have to assign responsibilities for creating the requirements specification at the beginning of a project.

Problem. On behalf of the mission, all stakeholders have to negotiate the requirements until they reach a mutual agreement, but the specialists concentrate on technical work and thus are difficult to access for negotiation. The

WHY READ THIS ARTICLE?

You can reinvent your RE process from scratch—with the risk of repeating previous failures—or you can construct it from known patterns. This article reflects on the nature of patterns and how to obtain them, and reports brief experiences of pattern use for specific RE practices. As such, it provides a starting point for greater sharing of good RE practices and experiences across our communities.

—Neil Maiden, Suzanne Robertson, and Christof Ebert, guest editors

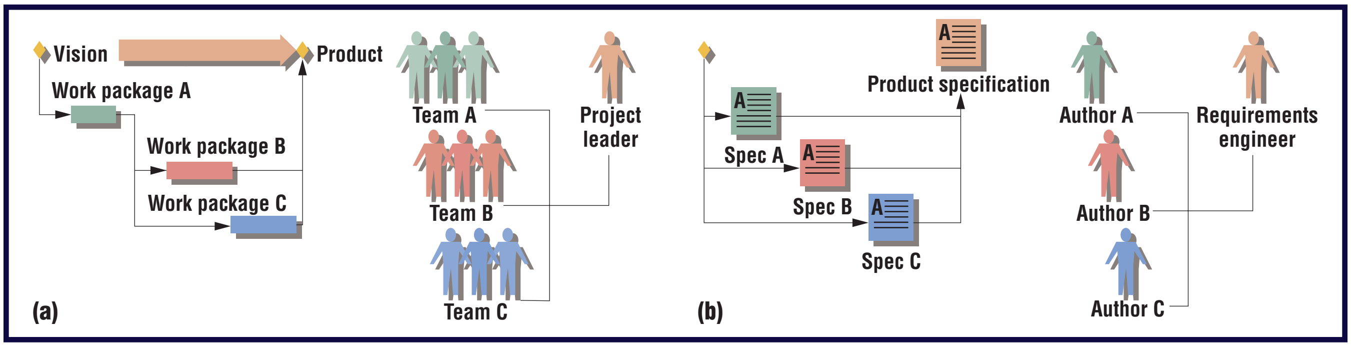


Figure 1. (a) Project structure and (b) specification procedure: Every team appoints an author who can contribute to the project specification according to the team's viewpoint. The requirements engineer coordinates the authors, and the project leader coordinates the project teams.

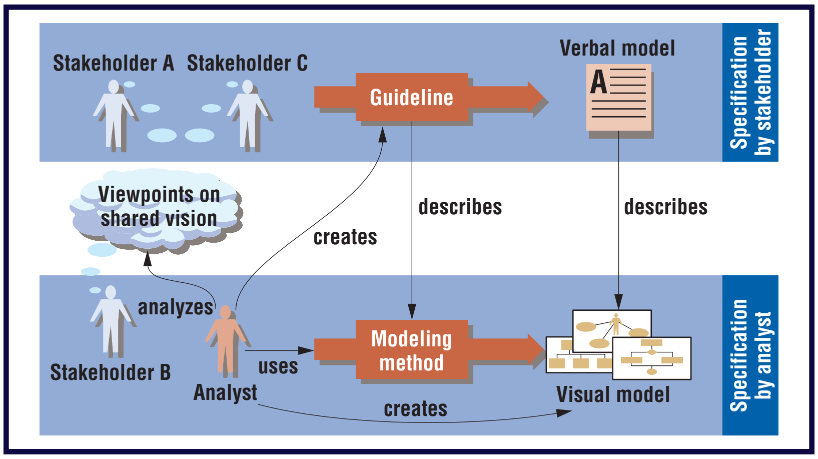


Figure 2. Guidelines enable stakeholders to create a project specification autonomously.

project leader has to efficiently organize collaboration and information exchange between the stakeholders.

Solution. Organize the specification procedure along the existing project structure. Figure 1 shows specification documents and their authors, which reflect the work packages and project teams. Every team has to appoint an author who can contribute to the project specification according to the team's viewpoint (see Figure 1b). A central requirements engineer coordinates the authors similar to how the project leader coordinates the project teams.

Application areas. The proposed organization has proven useful in different kinds of distributed projects, such as

- Locally distributed projects, in which the teams are spread across different locations
- Projects with dissimilar stakeholders or expert groups originating from different disciplines or cultural backgrounds
- Projects with large fluctuations in the team over time

Consequences. One positive effect is that the resulting specification describes a vision from different perspectives and is balanced in granularity. Another positive consequence is that the requirements engineer's role ensures a moderated and formally documented negotiation procedure.

Examples. In project #1, the project leader organized the teams according to plant components and different engineering disciplines. For example, there were groups for cryogenic plants and tunnels as well as for survey or electric supplies. Project #2 was structured along business processes with one responsible person per process and an additional team for the system architecture.

Experience. The requirements specifications should be made available to the entire project team—for example, by using a requirements management system. Also, team members should use role playing to represent stakeholders who can't access the project.

RE Pattern 2: Create a Specification Guideline by Tracking How an Analyst Works

This pattern addresses the need to develop and provide a specification guideline.

Objective. Enable stakeholders to make complete, balanced, and consistent contributions to the project specification.

Context. In a distributed project, independent teams have to envision and elicit requirements.

Problem. Domain experts can't delegate the writing of requirements specifications; they have to write technical specifications personally, but they lack the knowledge of the appropriate methods and tools.

Solution. Provide a specification guideline that lets the stakeholders build a verbal specification similar to how an analyst would build a visual model and of similar quality (see Figure 2). Track the steps of an analyst creating a business model and convert them into a guideline for the project team.

Application areas. Specification guidelines have been successfully used by

- Teams in long-term projects with frequent changes
- Expert teams in need of highly specialized knowledge for specifications
- Stakeholder groups of loosely coupled individuals who aren't available for other elicitation methods

Consequences. One positive effect is that guidelines enable remote specification work, thus supporting distributed projects. However, a drawback is that deficiencies in guidelines lead to comparable deficiencies in specifications.

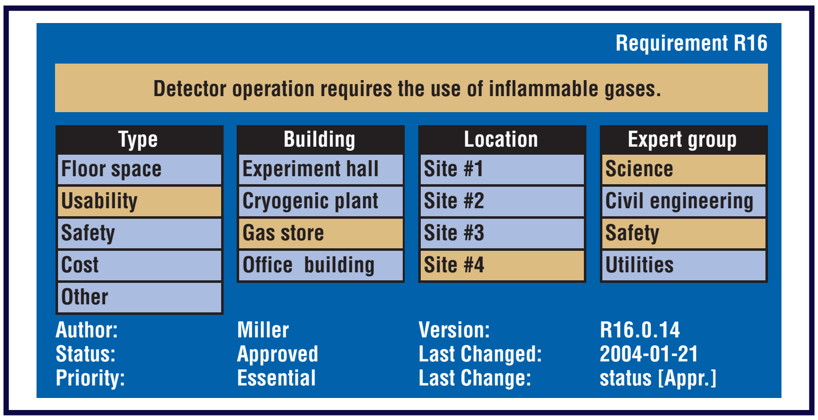
Examples. Analysts developed the guideline for project #1 following a general scheme for creating object-oriented models. It iteratively asked for the intended functionality of plant components, their required equipment and resources, and their operational procedures. Project #2 created similar guidelines, which asked user groups for their specific tasks, the information they require or produce when performing such tasks, and how they enter or display that information in forms or reports.

Experience. Because the guidelines can be used in early project phases, when model granularity is still coarse, we were able to produce, with reasonable effort, efficient guidelines that deliver fast results to users. In all the projects, the analysts have translated some of the specifications back into UML models to control the specification quality and improve the guidelines.

RE Pattern 3: Use Requirements Index Cards

This pattern describes a straight-forward specification format.

Objective. Make specifications from stakeholders with different cultural backgrounds compatible in format and detail level as a basis for further analysis.



Requirement R16

Detector operation requires the use of inflammable gases.

Type	Building	Location	Expert group
Floor space	Experiment hall	Site #1	Science
Usability	Cryogenic plant	Site #2	Civil engineering
Safety	Gas store	Site #3	Safety
Cost	Office building	Site #4	Utilities
Other			

Author:	Miller	Version:	R16.0.14
Status:	Approved	Last Changed:	2004-01-21
Priority:	Essential	Last Change:	status [Appr.]

Figure 3. Requirements index card.

Context. The requirements engineer has to analyze specifications for completeness and consistency.

Problem. For efficient comparison and correlation, project teams have to specify requirements in a unique, preferably text-based format, yet different teams prefer their familiar individual formats.

Solution. Use the metaphor of index cards for requirements. Each requirement index card contains a specification statement plus additional attributes for classification. Include general attributes for requirements management and to track project progress. Include customized attributes such as the affected system component and the relevant expert groups for filtering and correlating requirements (see Figure 3).

Application areas. We've observed benefits in

- Interdisciplinary projects with different established "cultures" for specification (for example, civil engineers are used to technical drawings, safety engineers prefer schematics, and IT staff prefer models) in which the stakeholders had to agree on a common format
- Projects with different interest groups, where the groups wanted to filter and access the requirements according to criteria relevant to them

Consequences. One positive effect is that classification helps to filter the specification and create domain-specific views. However, vague attributes or attributes with overlapping semantics might lead to useless classification.

Examples. In project #1, the team used index

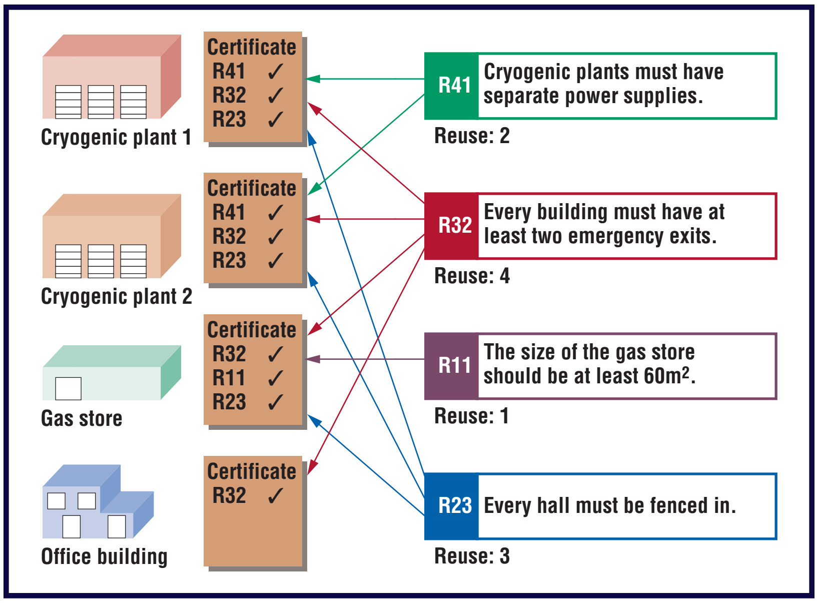

Cryogenic plant 1
Certificate R41 ✓ R32 ✓ R23 ✓
Cryogenic plant 2
Certificate R41 ✓ R32 ✓ R23 ✓
Gas store
Certificate R32 ✓ R11 ✓ R23 ✓
Office building
Certificate R32 ✓
R41 Cryogenic plants must have separate power supplies.
Reuse: 2
R32 Every building must have at least two emergency exits.
Reuse: 4
R11 The size of the gas store should be at least 60m2.
Reuse: 1
R23 Every hall must be fenced in.
Reuse: 3

Figure 4. Certificates for requirements approval.

cards with several domain-specific attributes for classification. The attributes included the affected plant components, the domains that the requirement affected, and the requirements type. In project #2, the cards included additional attributes for contracts (such as vendor estimates of the maximum cost for realizing requirements) and for relating functional requirements to the work packages that the team needed to perform to realize those requirements.

Experience. The required attributes were hard to guess in the beginning but quickly emerged as different users started accessing the specification. The users could represent a set of index cards as a table with rows corresponding to requirements and columns to their attributes and hence could easily implement the set using lightweight office products.

RE Pattern 4: Generate Approval Checklists

This pattern addresses the verification of requirements fulfillment.

Objective. Ensure that the developers have created a given version of a product according to the agreed-upon version of the specification.

Context. The team leaders have to check a product version to ensure that it complies with its requirements specification.

Problem. The requirements engineer can't keep a record of requirements fulfillment by simply check marking the requirement, because the specification reuses requirements for different components.

Solution. Use dynamically generated checklists that list contributing requirements for approval items (see Figure 4). Generate the checklists by querying the requirements database, and add fulfillment information. Use the complete checklist document as an approval certificate.

Application areas. The checklist approach benefits any project that reuses the same requirements on several occasions—for example, projects that

- Follow an incremental process model, delivering products in versions and using reiterated approvals
- Deliver product families

Consequences. One positive effect is that approval certificates let the requirements engineer perform product and requirement versioning and track approval history. Another advantage is that certificates can be formally treated as independent documents—reviewed, released, published, and so forth—because they decouple approvals from the specification. However, a drawback is that, in some cases, test users can't recognize whether a requirement is satisfied directly from the requirement and thus demand further instructions (such as a test case).

Examples. In project #1, the team used generated approval certificates to review CAD design models of several comparable buildings to ensure that they could accommodate the intended installations. Project #2 used generated checklists for work packages (instead of product versions) to incrementally check which requirements were completed.

Experience. Filtering for the system component to be approved yields a complete certificate, which the requirements engineer can then further refine by filtering out, for example, affected domains or stakeholder groups.

Pattern analysis

We now investigate the common structure of the patterns we've presented.

What makes a pattern a pattern?

We base our discussion on "the window place," a frequently quoted pattern from Alexander's *The Timeless Way of Building*.[1] When designing a comfortable living room, Alexander recommends placing a sitting area close to the windows. In rooms where this isn't the case, people are caught in a conflict: they're drawn to the chairs to sit down and relax but also to the windows where the light is. Using the window place pattern resolves this conflict (see Figure 5a). In other words, patterns help prevent or reduce stressful situations—stress in this context being difficulties that cause worry or emotional tension.

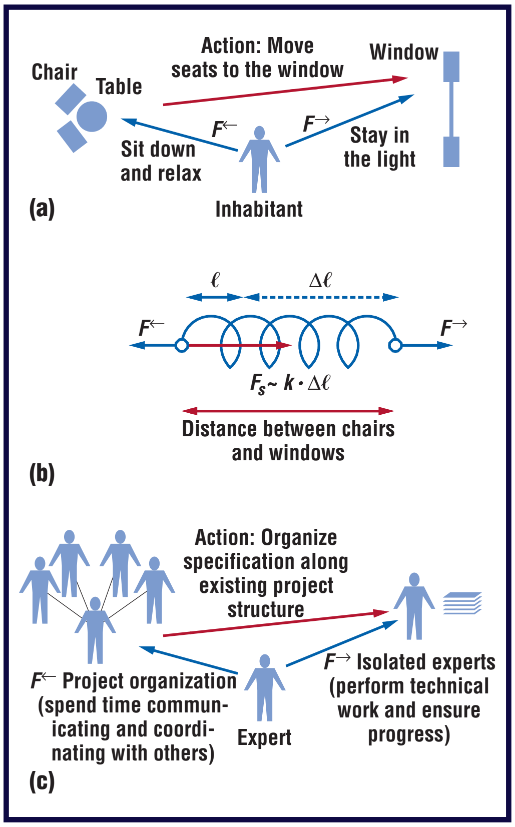
We now draw an analogy from a physics definition of stress: a pressure or force that produces strain on a physical body. We can illustrate stress using, for example, a spring of length ℓ and two forces $F^{\leftarrow}$ and $F^{\rightarrow}$ that are applied to its two ends (Figure 5b). The spring will stretch by a certain distance $\Delta\ell$ and react with a stress force $F_s = k\,\Delta\ell$ (k is a constant describing the spring). With some imagination, you can see the spring corresponding to the person in the window place pattern. Being comfortable corresponds to the spring being relaxed—in other words, $F_s \sim 0$. $F^{\leftarrow}$ and $F^{\rightarrow}$ stand for the forces pulling the person toward the chairs and window, respectively, and $\Delta\ell$ measures the stress as (literally) proportional to the distance between the window and chairs. A pattern would simply recommend keeping $\Delta\ell \sim 0$ to achieve $F_s \sim 0$.

When we take this discussion back to the window place pattern, a quality goal ($F_s \sim 0$) and two forces ($F^{\leftarrow}$ and $F^{\rightarrow}$) pulling in opposite directions describe the window place's stress situation, and the pattern offers a resolution. Generalizing this idea, we suggest that a pattern P is a quadruple, $P = (T, F^{\leftarrow}, F^{\rightarrow}, A)$, where T is a task with a quality goal, $F^{\leftarrow}$ and $F^{\rightarrow}$ are opposite forces generating a stress force, and A is an action compensating for the difference between the forces.

Revisiting the RE patterns

We can verify the approach's applicability by taking RE pattern #1 as an example and looking at the observations reported by the involved stakeholders. The reports might sound different from different viewpoints. For example, domain experts might explain how they're being torn apart when addressing their task, saying something like

> *When I have to create an advanced technical specification in a distributed interdisciplinary environment, I have to spend time not only doing technical work but also communicating and coordinating with the project team.*



Chair
Table
Action: Move seats to the window
Window
Sit down and relax
F←
F→
Stay in the light
Inhabitant
(a)
ℓ
Δℓ
F←
F→
Fs~ k · Δℓ
Distance between chairs and windows
(b)
Action: Organize specification along existing project structure
F← Project organization (spend time communicating and coordinating with others)
Expert
F→ Isolated experts (perform technical work and ensure progress)
(c)

Figure 5. (a) Analyzing a "window place" using (b) a physics analogy that we then (c) apply to RE pattern #1. ($F^{\leftarrow}$ and $F^{\rightarrow}$ are conflicting forces.)

Project leaders might report on or instruct the usage of the pattern as follows:

> *In distributed projects that contain several specialized expert groups, I would organize the specification procedure along the project organization. Using established communication channels reduces the communication overhead and leaves time for technical work.*

Outside observers would report the successful application of the measure when solving a problem:

> *In an interdisciplinary plant construction project, distributed expert groups need to create common planning documents. Initially, the groups were rather self-contained and too focused on their responsibility. After organizing the specification procedure along the existing project organization and using the established communication channels, communication and specification quality improved.*

Now we extract the common facts from the observations using the proposed pattern structure:

- T: Negotiate mutually agreed specification

Table 1

RE patterns

Pattern	Task	Quality goal	Conflict	$F^{\leftarrow}$	$F^{\rightarrow}$	Action
Organize the Specification along the Project Structure	Create specification	Mutual agreement of stakeholders	Communication, information exchange	Requirements have to be negotiated until a mutual agreement has been reached	Specialists concentrate on technical work and thus are difficult to access for negotiation	Organize specification along existing project structure
Create a Specification Guideline by Tracking How an Analyst Works	Elicit requirements	Completeness and consistency	Skill level	Creating specifications requires specialist knowledge in methods and notations	Writing specifications requires expert knowledge in the stakeholder domains	Provide guideline that enables inexperienced stakeholders to write requirements
Use Requirements Index Cards	Write requirements	Compatibility and good structure	Diversity of formats	Requirements have to be specified in a unique, preferably text-based format	Domain experts apply individual documentation and specification formats	Use (electronic) index cards to capture requirements with attributes
Generate Approval Checklists	Verify and approve product	Version tracking	Degree of reuse	Stakeholders expect lists of requirements as checklists for approvals	Requirements apply to several components and hence can't be marked "approved" directly	Create checklists dynamically by filtering the requirements

- $F^{\leftarrow}$: Requirements have to be negotiated until a mutual agreement has been reached
- $F^{\rightarrow}$: Specialists concentrate on technical work and thus are difficult to access for negotiation
- *A*: Organize specification structure along existing project organization

We see that the stress force arises with (lack of) communication—more precisely, with the low intensity of the project team's information exchange (see Figure 5).

Table 1 presents our four RE patterns in the proposed pattern format. The conflicting forces are sometimes abstract, but we find this appropriate and expected in engineering discussions. Generally, the proposed structure helps clarify the problem that a pattern addresses, but the analogy with forces—especially for computing the stress level—shouldn't be taken too literally.

Pattern mining: Making patterns accessible

We can use the pattern structure to create indexes, thus enabling pattern retrieval based on, for example, selected RE tasks (*T*) or specific project conditions ($F^{\leftarrow}$, $F^{\rightarrow}$). We've started to create a pattern database to

- Provide an easy way to access the pattern collection
- Support the discovery of patterns in a collection of observations

Figure 6a shows a simplified sketch of our database schema. We can enter observations into the database, formatted as discussed earlier. The database establishes patterns by querying for two or more observations with identical combinations of forces and action. We can then further elaborate the descriptions of the discovered patterns. Figure 6b shows a report from a test database containing the patterns described in this article. Currently, we're implementing a Web front end to the database to create an online pattern repository (see http://repare.desy.de).

Experience

The projects successfully adopted RE using the presented patterns, and the analysis method we developed while introducing the patterns has since been successfully applied to other projects.[9]

Using the patterns

The patterns we presented helped practitioners discover potential specification conflicts, eliminate possible project risks, and clarify responsibilities. Explaining the patterns, especially the conflicting forces and their resolution, increased RE acceptance. Furthermore, promoting the patterns close to project milestones ensured better management support. At these times, RE became especially visible when providing the essential inputs for validating approvals and contracts.

Additionally, patterns let the teams introduce RE in phases: first, by encouraging users to write specifications with office tools, then by letting them categorize the requirements, and finally by letting them use queries and traces for approvals. This gave the teams sufficient time to fully adopt RE and let them adjust the speed of introduction to their learning curve.

Also, it's interesting to note that the patterns for RE activities were observable and applicable beyond software engineering and computing. In project #1, they applied to mechanical and civil engineering for plant construction.

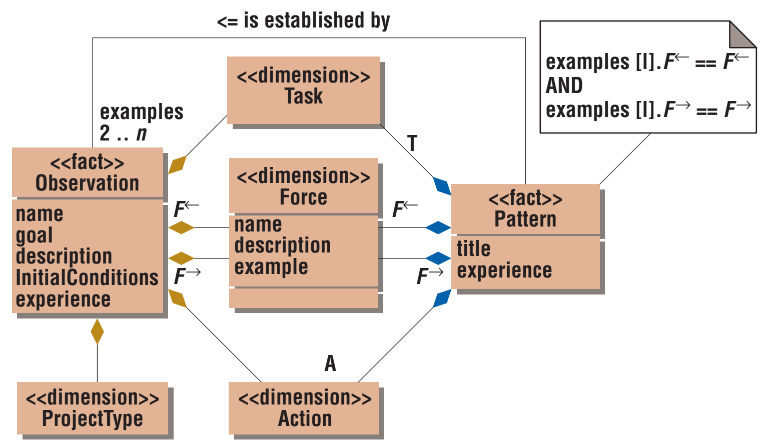
Pattern analysis

The proposed pattern structure helped users decide which observations qualified as pattern candidates. For example, templates for specification documents do not solve a conflict in the sense of this article, so we considered them instructions or best practices and didn't include them in further analysis.

The pattern format was also helpful when analyzing project situations by drawing attention to the real issue. We repeatedly observed that project leaders were initially not addressing the appropriate conflict when making observations. For example, in project #1, the distribution of the teams was initially considered more critical than the low rate of information exchange.

The proposed pattern structure led to a natural way of sharing and exchanging experience. When project teams provided narrative observations from their work, they could categorize the observation records, add them to a repository, and compare and relate them to other observations, thus identifying patterns of successful RE activities.

The patterns we presented here were an initial attempt to capture engineering knowledge in an accessible format. The German Informatics Society has established a working group to collect and analyze observations on a broader scale to reveal additional patterns for RE activities.[9]



(a)

IF	requirements have to be specified in a unique, preferably text-based format
BUT	domain experts apply individual documentation and specification formats
THEN	use requirements index cards
TO	write requirements which are compatible in format

(b)

Figure 6. (a) Schema for pattern mining and (b) sample database output.

References

1. C. Alexander, *The Timeless Way of Building*, Oxford Univ. Press, 1979.
2. C. Alexander et al., *A Pattern Language*, Oxford Univ. Press, 1977.
3. E. Gamma et al., *Design Patterns*, Addison-Wesley, 1994.
4. F. Buschmann et al., *Pattern-Oriented Software Architecture, Vol. 1: A System of Patterns*, John Wiley & Sons, 1996.
5. H.-E. Eriksson and M. Penker, *Business Modeling with UML—Business Patterns at Work*, John Wiley & Sons, 2000.
6. I. Sommerville and P. Sawyer, *Requirements Engineering—A Good Practice Guide*, John Wiley & Sons, 1997.
7. L. Hagge et al., "Integrated Information Management for TESLA," *Proc. Computing in High Energy and Nuclear Physics* (CHEP 03), 2003; http://arxiv.org/ftp/cs/papers/0306/0306079.pdf.
8. L. Hagge and K. Lappe, "Patterns for the RE Process," *Proc. IEEE Joint Int'l Conf. Requirements Eng.* (RE 04), IEEE CS Press, 2004, pp. 90–99.
9. K. Lappe et al., *Requirements Engineering Patterns—An Approach to Capturing and Exchanging Requirements Engineering Experience*, DESY 04–223, DESY, 2004.

About the Authors

Lars Hagge is a senior scientist at Deutsches Elektronen-Synchrotron in Hamburg, Germany. He heads the information management department, which is supporting process optimization and providing information systems for lifecycle management. He received his PhD in physics from the University of Hamburg and is an active member of the German Informatics Society. Contact him at Deutsches Elektronen-Synchtrotron, IPP, Notkestr. 85, D-22607 Hamburg, Germany; lars.hagge@desy.de.

Kathrin Lappe is an engineer in the information management department at Deutsches Elektronen-Synchrotron. She is currently working as a requirements engineer in a plant construction project. She holds a diploma in library and information science from the University of Applied Sciences in Hamburg. She is spokesperson of the RE patterns working group of the German Informatics Society. Contact her at Deutsches Elektronen-Synchtrotron, IPP, Notkestr. 85, D-22607 Hamburg, Germany; kathrin.lappe@desy.de.