Deliverable reference:	Date:	Responsible partner:
D06.1	27 January 2012	SINTEF

Bridging Resources and Agencies in Large-Scale Emergency Management

BRIDGE is a collaborative project co-funded by the European Commission within the Seventh Framework Programme (FP7-SEC-2010-1)
SEC-2010.4.2-1: Interoperability of data, systems, tools and equipment
Grant Agreement No.: 261817
Duration: 1 April 2011 – 31 March 2015

www.sec-bridge.eu

Title:

BRIDGE Design Pattern Library

Editor(s):
Ragnhild Halvorsrud

Approved by:
Andreas Zimmermann

Classification:
Public

Abstract / Executive summary:

Deliverable D06.1 is a Design Pattern Library to be used in the BRIDGE project.
The essence of this deliverable is to provide a framework that enables collaborative insertion, discussion and sharing of design patterns in the context of the BRIDGE project. The idea is to populate the library with patterns that emerge – internally and externally - during the project lifetime.

Please visit the library at http://pattern-library.sec-bridge.eu/

This report provides background knowledge for using the BRIDGE Design Pattern Library. A general introduction to design patterns is presented, and the use of patterns in the context of emergency management. Furthermore, it describes the use of patterns in the context of the BRIDGE project. Scope and sources of patterns are described, as well as the implementation of the library, together with its features and functionality. A literature review is also provided, demonstrating a number of design principles and user interfaces in the field of emergency management.

Document URL:
http://www.sec-bridge.eu/deliverables/...

ISBN number:
978-82-14-05402-6

Table of Contents

Version History

Version	Description	Date	Status
1	Creation of document, table of contents,	Oct 11 2011	Completed
2	First version of web site launched, report chapters in place	Nov 15 2011	Completed
3	Adjustments and updates to harmonize report and web site	Nov 24 2011	Completed
4	Ready for review	Nov 28 2011	Completed
5	Integrated comments from 1st review	Dec 16 2011	Completed
6	Integrated comments from 2nd review	Dec 22 2011	Completed
7	Integrated comments form technical coordinator	Jan 20 2012	Completed
8	Released version	Jan 27 2012	

Contributing partners

SINTEF Information and Communication Technology (ICT)

Forskningsveien 1
N-0373 Oslo
Norway

Ragnhild Halvorsrud
ragnhild.halvorsrud@sintef.no

Aslak Wegner Eide
aslak.eide@sintef.no

University of Klagenfurt

Alpen-Adria-Universität Klagenfurt
Universitätsstrasse 65-67
9020 Klagenfurt
Austria

Daniela Pohl
Daniela.pohl@uni-klu.ac.at

Fraunhofer Institute for Applied Information Technology (FIT)

User-Centered Ubiquitous Computing Department, Schloss Birlinghoven
53754 Sankt Augustin
Germany

René Reiners
rene.reiners@fit.fraunhofer.de

List of Figures

1 Introduction and overview

1.1 The BRIDGE project

The overall goal of BRIDGE is to increase safety of citizens by developing technical and organisational solutions that significantly improve crisis and emergency management. BRIDGE will build a system to support interoperability – both technical and social – in large-scale emergency management. The system will serve as a bridge between multiple First Responder organisations in Europe, contributing to an effective and efficient response to natural catastrophes, technological disasters, and large-scale terrorist attacks.

One of the main pillars in the BRIDGE project is the development of advanced human-computer interaction techniques, in particular improvements in decision support systems, tools and equipment for emergency responders.

Please visit the BRIDGE website for more information: http://www.bridgeproject.eu

1.2 About this deliverable

The D06.1 Design Pattern Library is the first deliverable in the work package *Interaction Design* (WP06). This work package comprises the development of common interface components both for advanced interaction and for visualization purposes. The content of the deliverable is described as *"an online library which will collect best practices based on our efforts in the project"*. Table 1 reveals the key information about D06.1 based on the BRIDGE Description of Work.

KEY INFORMATION	
Work package:	WP06 Interaction design
Deliverable title:	D06.1 BRIDGE Design Pattern Library
Deliverable short name:	BRIDGE DPL
Deliverable description:	An online pattern library which will collect best practices based on our efforts in the project.
Delivery month:	M9 (December 2011)
Delivery date:	Jan 1, 2012
Nature of deliverable:	O (other)
Dissemination level:	PU (public)
Responsible partners:	SINTEF, University of Klagenfurt, Fraunhofer FIT

Table 1 Key information about this deliverable

The BRIDGE Design Pattern Library (DPL) is implemented as a combination of three components, of which the online library is regarded as the main component (Figure 1). The online library has been set up to host a collection of design patterns developed during the course of the project. This report provides background knowledge for using the online library, and documents its features and functionality. In addition, a Mendeley literature repository is used for easy managing and effective sharing of research papers concerning design patterns in the domain of crisis management.

The essence of this deliverable is to provide a framework that enables collaborative insertion, discussion and sharing of design patterns in the context of the BRIDGE project. The idea is to populate the library with patterns that emerge – internally and externally - during the project lifetime. A few example design patterns from the crisis management domain are available in the BRIDGE DPL, to illustrate functional and graphical aspects of the library.

Figure 1 Overview of Deliverable D06.1

The BRIDGE Design Pattern Library is available at: http://pattern-library.sec-bridge.eu/ and will be linked from the BRIDGE project site.

The literature repository about design patterns is available here:
http://www.mendeley.com/groups/1414573/design-patterns-and-pattern-languages/

1.3 Overview of report

Chapter 2 provides a general introduction to the design patterns. It also introduces design patterns in the context of the emergency management domain. An overview of online pattern collections is also provided here.

Chapter 3 describes the use of design patterns in the context of the BRIDGE project. It defines the scope of the library and discusses possible sources of patterns. This chapter also describes how we involved the BRIDGE participants in developing the library, and finally how the library was implemented.

Chapter 4 provides a functional description of the BRIDGE Design Pattern Library. It covers functionality, roles, pattern attributes, pattern submission, pattern life-cycle and a discussion of possible future extensions.

Chapter 5 provides a visual walk-through of the BRIDGE Design Pattern Library.

Chapter 6 summarizes state-of-the-art design in emergency management practice. A number of design principles and user interfaces from a literature review are presented.

2 Introduction to design patterns

This chapter provides a general introduction to design patterns and the various types of patterns in the field of Human Computer Interaction (HCI). Design patterns in the context of emergency management are also introduced, together with an overview of inspirational pattern collections.

2.1 What are design patterns?

Design patterns are used to describe best practices and effective design solutions, and for capturing and sharing design knowledge with other people faced with the same problem and context. The solution proposed by a design pattern should be generic rather than specific, so that it can be implemented in numerous different ways.

Design patterns originated as an architectural concept introduced in the seminal book *"A Pattern Language"* from 1977, written by Christopher Alexander (1977). In this pioneering work, Alexander describes design patterns as elements of an architectural language, defined in the following way:

> *"The elements of this language are entities called patterns. Each pattern describes a problem that occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing it the same way twice."*

Seeing how design patterns could also be useful in other contexts, the work of Alexander inspired other engineering disciplines to adopt the concept. In particular, the famous Gang of Four extended the notion of patterns into the domain of software engineering, and constructed 23 classic software design patterns (Gamma, et al., 1994). Since then, the concept of design patterns also became essential in the HCI domain, where patterns are commonly used to describe and preserve solutions to recurring UI design problems.

The book "Designing Interfaces" describes the influence of patterns in user interface design (Tidwell, 2010). Patterns are described as "best-practice within a given design domain" for improving objects, like user interfaces. As Alexander and Tidwell mention in their work, each implementation of the same pattern differs somehow in its characteristics although it comes from the same origin. Patterns should be seen as description of a problem solution as starting-point and not as fixed design rules. Patterns are therefore defined with the following characteristics (Tidwell, 2010):

- Concrete, not general
- Valid across different platforms and systems
- Products, not processes
- Suggestions, not requirements
- Relationships among elements, not single elements
- Customized to each design context

Within user interface design, design patterns are general solutions to recurring problems that enable users to use a system in a more practicable and easy manner. This especially is a concern in emergency management systems as people must feel confident with the system they use under a stressful situation.

In terms of content, a design pattern should always include a description of the context where the pattern is relevant, a description of the problem being addressed, and a description of the solution to this problem. It is important that a design pattern has a clear and descriptive name, so that it can easily be identified by others, and perhaps more importantly, an illustration that

demonstrates the solution proposed by the pattern. To exemplify the use of these fields, we have included a well-known web design pattern[1] in Table 2.

DESIGN PATTERN ATTRIBUTES	
Name:	LanguageSelector
Context:	On a multinational website, the users will require the content to be available in different languages.
Problem:	The users of a multinational website need a simple way of choosing their preferred language on the website.
Solution:	Enable the users to select their language through a box where they can click on the flag of their respective nationality, or by selecting their nationality from a pull-down-menu.
Illustration:	

Table 2: Design pattern example

The benefit of using patterns is that they communicate insights into common design problems and reflect solutions that a community has developed over time. A given pattern is not always the optimal solution in every case, but tends to work in practice. A user interface with bad design results in resistance when using a software system. Therefore, the description of *anti-patterns* is also a valid information source for UI designers. They can be commonly used but turns out to be ineffective or counterproductive in practice. Anti-patterns can be seen as a "DON'T rule for design" (Reiners, Astrova, & Zimmermann, 2011).

2.2 Classes of design patterns

The usefulness of design patterns has been recognized within the field of HCI, taking the form of a variety of pattern sub-types, such as: *User interface design patterns*, *Interaction design patterns*, *User experience patterns*, *Usability patterns*, *Web design patterns*, and *Visualization patterns.* However, as highlighted by Stiso (2011), there seem to be a great overlap between the different pattern types, and generally a lack of consensus regarding how they differ from one another.

In a review of design for safety-critical systems, Grill and Blauhut (2008) proposed the following categorization of patterns: 1. Interaction design patterns 2. User interface design patterns 3. Information visualization patterns. To decide which pattern types to use in the BRIDGE DPL, online pattern collections and relevant literature has been reviewed. Inspired by Tidwell (2010) and Grill & Blauhut (2008), the following categorization was chosen for BRIDGE (see Figure 2):

- **UI Patterns** and **Visualization Patterns** address graphical representations of visual elements and components in a user interface. The two variants often overlap, and is therefore put into a common category.
- **Interaction Patterns** describe interactive elements in a specific context. They make use of UI or Visualization patterns, or perhaps a combination of them.
- **Application Patterns** describe solutions for an application concept as a whole, the context of an application, and higher-level concepts of an application.

[1] Variants of this pattern can be found in several online pattern collections, see references in Section 2.4.

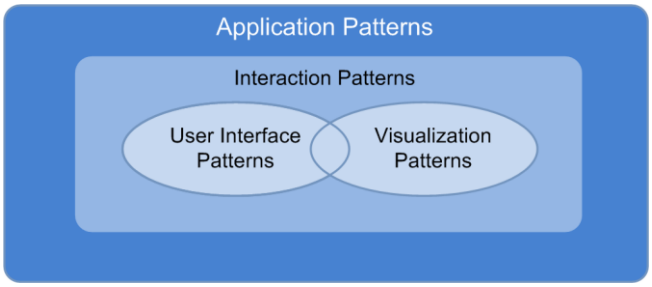


Figure 2 Categories of design patterns

The innermost category (UI and visualization patterns) represents the low-level and mostly static elements forming the building blocks for the next level. In contrast, the interaction patterns are associated with a dynamic element(s) in response to a user's action and manipulation. Application patterns are high-level composite constructs that encompass a number of UI and interaction elements.

2.3 Design patterns in emergency management

Designing interfaces within the area of emergency management requires special care, since the end users might be in a stressful situation where lives and values are at stake. General 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.

Grill and Blauhut (2008) describes a framework for identifying and validating design patterns within the field of emergency management through a user-centred design process. The process also facilitates creation, update and reuse of already known patterns; see the diagram in Figure 3. They extend the user interface design process beyond the traditional ones by introducing periodic evaluations of the pattern validity.

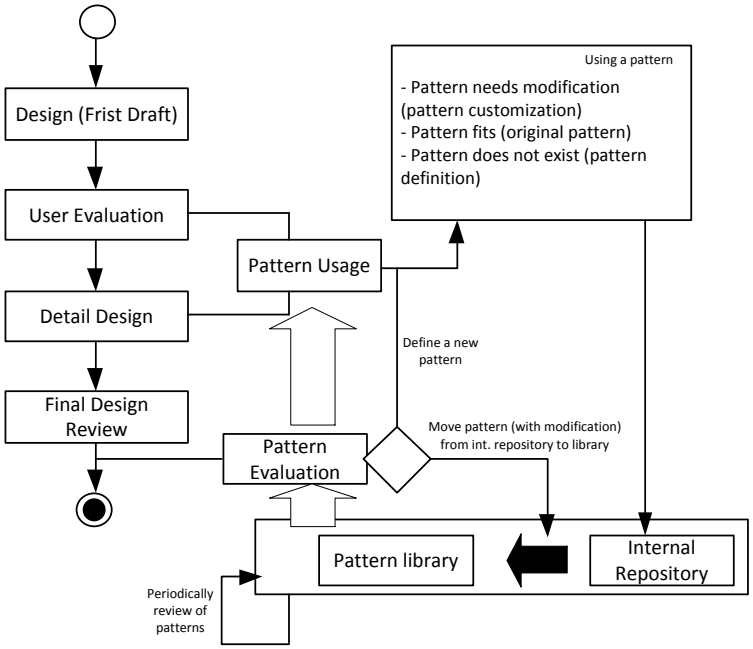


Figure 3: The Frequentis user interface design process for creation, reuse and update of design patterns (adopted from Grill & Blauhut, 2008)

The process starts with the creation of an early draft of the UI. This step is based on a previous workshop, held to analyze the requirements with the user. During the latest design phase, the design is evaluated with the users based on their expectation. The process therefore results in an iterative manner to have an ongoing improvement of the UI until no more changes are needed and the UI design process freezes. Within the design phase, design experts have the possibility to access a pattern library, containing important patterns identified from previous projects. The pattern library can be browsed by the experts and the results are analyzed. As a result of the analysis new patterns are created if no existing pattern fits, updated, if the existing pattern can be more generalized, and reused, if the pattern fits the expectation of the experts. This results in an up-to-date pattern library containing the experiences of different UI designs in safety critical environments.

The pattern form used by Grill and Blauhut is, like most others, a modified form of the original work of Alexander (1977), see Table 3. A pattern map is used for browsing the Frequentis pattern library. The map is based on the pattern categorization (Interaction, UI design and visualization) described previously, making it easier to navigate and get the overview of the content.

DESIGN PATTERN ATTRIBUTES	
Name	A name identifying the pattern
Pattern Category	Categories defining the application area of the pattern
Illustration	Visualization of the problem
Problem	Problem statement
Context	How and where the problem occurs, and when the solution works
Forces	Reason for this design, challenges of the problem
Solution	Solution for the described problem
Rationale (optional)	Explanation for justifying the solution
Related Patterns	Similar patterns
Known Uses/Examples	Three or more independent instances of "real world" success
Notes (optional)	Additional notes

Table 3 Design pattern form (adopted from Grill & Blauhut, 2008)

2.4 References to online collections of design patterns

Most online pattern collections contain general design patterns for UI and interaction design – typically aimed towards web design. For use in emergency management the patterns must be carefully incorporated and validated according to the user and the context.

Yahoo! Design Pattern Library	http://developer.yahoo.com/ypatterns/
UI Patterns	http://ui-patterns.com/patterns
Welie.com Pattern Library	http://www.welie.com/
Interface Design Patterns	http://www.patternbrowser.org/
Patternry	http://patternry.com/patterns/
Quince Pattern Library	http://quince.infragistics.com/
HCIpatterns.org	http://www.hcipatterns.org/patterns
The Design of Sites:	http://www.designofsites.com/design-patterns/
Designing Web Sites:	http://designingwebinterfaces.com/explore

Endeca Pattern Site:	http://patterns.endeca.com/content/library/en/home.html
Android Patterns	http://www.androidpatterns.com/
Mobile UI Patterns:	http://mobile-patterns.com/
UI Anti Pattern Library:	http://ui-patterns.com/blog/User-Interface-AntiPatterns
Android Anti-Pattern:	http://www.androiduipatterns.com/p/android-ui-pattern-collection.html

3 Scope, sources and implementation

This chapter discusses the use of design patterns in the context of the BRIDGE project. It defines the scope of the library and discusses possible sources of patterns. The development of the library is also discussed, both on a conceptual and technical level.

3.1 Scope

One of the main pillars in the BRIDGE project is the development of advanced human-computer interaction techniques, in particular improvements in decision support systems, tools and equipment for emergency responders. Several design concepts, user interfaces, interactive elements and applications will be developed and evaluated during the course of the project.

The BRIDGE DPL deliverable is described as "an online library which will collect best practices based on our efforts in the project". This signals an acceptance of design elements in the library in a less strict fashion than the more traditional interpretation of design patterns. It is likely that the insights will take the form of conceptual design principles and examples that have been validated by means of case studies, demonstration activities and validation processes within the course of the project. Hence, it might be challenging to argue that the validated design solutions can be regarded as design patterns in the traditional sense, or that their applicability extends beyond the context of BRIDGE.

Thus, the BRIDGE DPL will apply a less strict definition of design patterns by including design concepts that are under consideration and most probably not the "optimal solution" to a problem. By applying this less strict definition, we expand the scope of the DPL content to include the insights gained from evaluating design solutions in workshops and case studies and demonstration activities with domain experts throughout the BRIDGE project. This "disclaimer" is also explained on the DPL website, making sure that the users of the library are informed.

The main content of the library will thus be a collection of design solutions and examples developed and validated within the BRIDGE project. Additionally, the library will contain design patterns originating from external sources, such as research within the field of crisis management systems, or user interface design.

Figure 4 illustrates the scope of content. BRIDGE DPL mainly concerns crisis/emergency management, but will also include elements of user interfaces and interaction that originates outside the emergency domain.

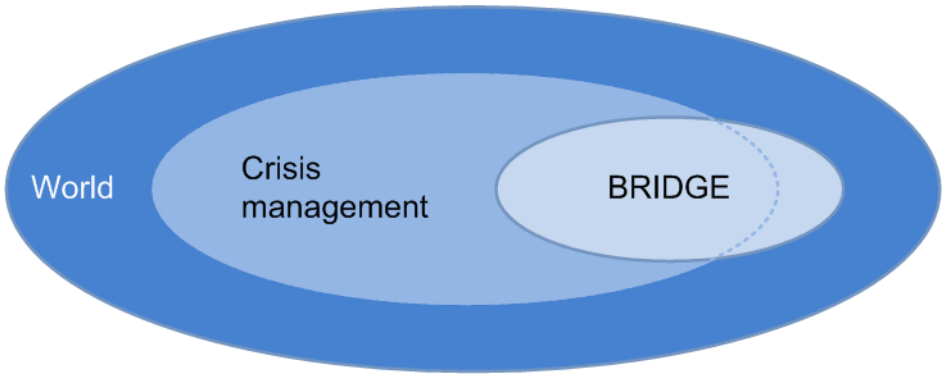


Figure 4: Pattern scopes of the DPL

Generally, the use and relevance of a pattern will vary depending on the context of use, the perspective of the user, and the mediating device. A variety of patterns should be expected as relevant, and DPL users can submit any type of pattern to the library. However, to maintain integrity of the library all submitted patterns will be subjected to a review process before

publication. This review process enables us to ascertain that the content of the library is well grounded and resented in a consistent way.

The BRIDGE DPL will serve as a public repository of best practices, where users can browse patterns, add content, and discuss design elements. As patterns may evolve and undergo updates throughout the lifetime of the project, the library should also support some notion of social collaboration, where users can discuss and give feedback on each other's content. External users of the library are encouraged to contribute, but will need to register for commenting on patterns, or contact the DPL team in order to submit patterns.

3.2 Sources and origin

An *internal* design pattern is derived either directly from the BRIDGE project, or from any of the project partners:

- **BRIDGE partners:** Earlier research conducted by the BRIDGE partners may include relevant and valuable insights that should be added to the library.
- **M. Sc. And PhD theses:** Theses written in relation to the BRIDGE project or other these work done for BRIDGE partners.
- **Workshops and case studies:** The BRIDGE project will arrange a number of workshops with domain experts, for early evaluation and co-design of prototypes and design concepts. These workshops will be an occasion for validating design patterns.

The BRIDGE DPL will also comprise design patterns originating from external sources[2] and contributors:

- **Generic patterns:** Online design pattern collections relevant for BRIDGE.
- **Patterns from related fields:** Patterns derived from fields not directly related to crisis management, such as warning design, risk perception and similar.
- **Literature and reports:** Patterns described in publications, reports and other literature sources (or guidelines and principles) that are relevant for BRIDGE.
- **Related EU projects:** Within the EU community, there are several projects that contain valuable insights relevant for BRIDGE (e.g. the COPE project).

Opportunities for patent and commercialization should also be considered before uploading internal contributions. However, it should be emphasized that a main goal of BRIDGE is to stimulate growth within European industry, and that dissemination of our results in public is a crucial part of achieving this goal.

3.3 Engaging the project partners

For the pattern library to become a useful tool within the BRIDGE project, it should comply with requirements and expectations of its end-users. From the early scoping process and through the implementation period, project members were encouraged to participate by contributing their ideas and pattern drafts, as well as providing feedback on the library content, pattern suggestions and early prototype.

Since many are not familiar with the notion of design patterns, an easy-to-read flyer was prepared to cover the basics of design patterns and their use (see Appendix 1). The flyer was disseminated to the project members to increase awareness about the library, along with an appeal to provide us with suggestions and feedback. The work in developing the BRIDGE DPL

[2] Material from external sources might be subject to copyright law, and should be handled accordingly. A reminder is provided in the online submission form.

was also presented during two consortium meetings (Delft, May 2011 and Bonn, Nov 2011). The purpose was to encourage and trigger contributions and content to the library.

To further increase the engagement level of the participants a temporal pattern submission form was published via the project collaboration portal, simulating a web form and enabling the project members to submit patterns. Regular e-mails were sent out, informing about the progress and prompting participants to submit pattern examples. The submission template can be found in Appendix 2. Several pattern suggestions (~10) were received[3] and later published through the BRIDGE DPL.

3.4 Implementation of the BRIDGE DPL

The scoping and planning of a design pattern library serving the BRIDGE project were described in the preceding section, and were based on analysis of existing approaches seen together with user requirements and ideas. The rules and processes that derived from the scoping process have been realized in a public, web based design pattern library. The BRIDGE DPL will be updated continuously, featuring necessary changes and updates as the pattern collection develops over time.

Since design patterns materialize as composite content embedded into a generally applicable template, an open source Content Management System (CMS) was chosen. Due to past experience with the system and the active community behind it, the Joomla! Content Management System[4] was chosen.

In the following, we would like to provide some arguments in favor of the choice for Joomla!

- Open Source Content Management Platform
- Large Supporting Community
- Easy management of content concerning publication status, publication and retraction date as well as authorship
- Dedicated roles and access control lists for items and processes (e.g., authors only write articles, publishers are responsible for putting them online and in place)
- Regular core framework updates concerning security and bugs (see below)
- Open for third-party extensions
- Mature framework relieving developers from standard programming tasks like login mechanisms, CRUD[5] database transactions for simple relationships, form handling and extension development
- Large number of freely and commercially available extensions to be used directly, or even extended due to their open source character

Joomla! is separated into two parts: A *back-end* side where administrators and authors manage the settings and installation or provide, edit and manage content, respectively. The *front-end* part of the platform displays content and prepares special views. Furthermore, the front-end handles login procedures for site visitors. An advantage of this approach is the distinction between the data and its representation that may come in different flavours. Thus, *templates* are assigned to the front-end and therefore prepare the data accordingly.

[3] by Nov 22, 2011

[4] http://joomla.org

[5] CRUD stands for Create, Read, Update and Delete

The Joomla! CMS has undergone many major development steps since its release in 2005. At the time of writing this deliverable the Joomla framework is available in version 1.7.3. Joomla! will most probably be updated to version 2.0 within the time frame of the BRIDGE project.

3.4.1 Extensions

The Joomla! CMS is not just capable of handling standard contents such as articles and web pages, it also provides standard extensions like, e.g., contact management, banners, a template switcher, multi-language support, menu management, access control lists, user management and login mechanisms together with content restrictions. Additionally, new (third-party) extensions can be installed via a built-in mechanism similar to an install wizard. This makes extension management very easy.

Joomla! supports various types of extensions, differing in complexity and purpose:

- *Components* represent the most complex extensions, providing configuration and item management in the back-end part of the Joomla! platform. For the front-end, components are linked and activated via menu items, enabling different views on the items of the component. For the standard installation these are articles that are displayed directly or summarized by category and section, respectively. In the context of BRIDGE patterns, user roles, devices and pattern states are the items to be managed in the back-end.
- *Extensions* are used to prepare component for special purposes in the front-end (e.g. menus and breadcrumbs). They can be placed at any position in the front-end, also as multiple instances.
- *Plugins* are used to prepare content before being displayed. For example, special HTML code or replacing of variables can be applied before providing the content.

3.4.2 Implementation of the design pattern management system

A special Joomla! component `com_patterns` was developed for managing the design patterns. Only dedicated editors and administrators of the back-end system have access rights to adjust these components. The BRIDGE DPL is publically available for everyone. User roles with appropriate access rights have been defined and implemented to ensure responsibility, quality and proper management of the BRIDGE DPL. The main interaction with the patterns like browsing, submission and commenting happens in the front-end. Mechanisms for browsing patterns by category or user role are also provided. Submission, however, is currently available for consortium members only. Everyone can comment and provide feedback, but "non-BRIDGERS" must register first. Please refer to Section 4.2 for details about roles and access rights. A walkthrough of the DPL front-end can be found in Chapter 5.

By default, a submitted pattern is not published until a DPL editor has reviewed the submission format and approved the fields provided by the author. If the current submission is not accepted, it is resent to the author for an update (see the diagram in Figure 5).

An additional module *mod_patterns* was implemented to display the latest changes in the library through a vivid sidebar next to the content on all front-end pages. This module provides a means to display patterns according to filters like e.g. *recently added, recently commented* and *recently approved*. The module is designed in a flexible way, allowing additional parameters to be added at a later stage.

3.4.3 Applied technology

Joomla! is implemented in PHP whereas the content data as well as configurations and settings for the installed extensions are stored in a MySQL database. PHP 5.3 and MySQL 5.1 are the minimum requirements for running the platform. GForge was used during the development,

providing an issue tracker, a wiki and a subversion repository management. The system is available at http://forge.fit.fraunhofer.de.

All technical documentation was placed in the wiki. This comprises documentation for local installations, tutorials and general information about developing under the Joomla! framework, and general information about the applied process. The issue tracker was used to keep track of open bugs, tasks and for managing the different software components' development status. The Subversion (SVN) repository supported the deployment strategy for the different DPL system versions.

3.4.4 Deployment strategy

The different versions of the BRIDGE DPL will be numbered according to the range of added features and reworking. Starting with version 1.0 (this deliverable), numbers in front of the dot l refer to major changes and the introduction of major features. The first number behind the dot will be used for regular updates and hotfixes (ranging from security updates in the Joomla! framework - to custom developments). The different deployment versions will also be tracked via the GForge issue tracker.

Working with Subversion, which classically divides the repository into the trunk, branches and tags, the main on-going development is kept in the trunk folder of the repository. Whenever the development has reached a stable state having assembled enough features for a new release, the current trunk is copied to a branch with the according release number. From then on, there is a development freeze meaning that only changes relevant for features of that version are committed to the branch. Eventual parallel development not belonging to the release will be committed continuously to the trunk. Regularly, the changes in the branch are merged with the trunk to ensure that the trunk remains at the latest state of development.

3.4.5 From design sketches towards the DPL platform

After the decision for the technical implementation of the DPL was made, early paper prototypes of the front-end layout were prepared and discussed. After several design iterations, involving project members outside the development team, a slim design suggestion was prepared, introducing the DPL to the consortium together with the flyer described in Appendix 2. A sample prototype is shown in Appendix 3 and was stepwise implemented per view and page.

4 The BRIDGE Design Pattern Library

4.1 Basic structure and functionality

As mentioned in Section 1.2 the essence of the deliverable is to provide a framework that enables a collaborative insertion, discussion and sharing of design patterns in the context of the BRIDGE project. The idea is to fill the library with patterns developed in the context of BRIDGE, and other relevant patterns derived from external sources.

The main functionality in the BRIDGE DPL is the following:

- *Browse* the online library
- *Submit* a new pattern to the library
- *Discuss* and comment on a design pattern
- Change the *status* of a design pattern (e.g. from "under consideration" to "approved")
- Provide *feedback* on the library and suggest improvements

These tasks will be described in more detail below.

4.2 Users, roles and access levels

The BRIDGE Design Pattern Library is open to the public – i.e. it can be reached and browsed by anyone having access to the Internet. Table 4 shows an overview of the different roles and user types associated with the DPL and their corresponding access rights and responsibilities.

ROLE	DESCRIPTION	DETAILS
BRIDGE external	Anyone - an arbitrary visitor to BRIDGE DPL	The following tasks are valid for all external users (as well as for all the other roles): • View patterns • Browse patterns External users must contact the DPL team for the following tasks: • Suggest new patterns • Comment and give feedback
BRIDGE internal	Any member of the BRIDGE project	• View and browse patterns • Submit patterns • Comment on patterns
Pattern responsible	Any member of the BRIDGE project who has submitted a pattern to the library	• The *pattern responsible* is the one who submitted the pattern in question. • The *pattern responsible* is responsible for maintaining and updating the pattern.
DPL editor	BRIDGE member assigned an editor role	• Responsible for quality assurance of submitted patterns • Can publish or reject design patterns
DPL validator	Any BRIDGE member involved in the validation of patterns	Participant in WP 9 (demonstration), WP 10 (validation), or otherwise involved in evaluation of patterns
DPL administrator DPL team	Administrator of DPL back-end	Back-end administrator with full access to modify the BRIDGE DPL site

Table 4 Roles associated with the BRIDGE DPL

Note that any visitor of the BRIDGE DPL can browse the design patterns, as well as contribute/suggest new patterns by contacting the DPL administrators. However, only BRIDGE members can submit patterns directly to the library and comment on patterns. Upon submission a user automatically becomes *pattern responsible* and thus responsible for maintenance (e.g.

upgrade) of the pattern in question. However, the *pattern responsible* is not necessarily the originator or the inventor of a pattern.

The role *DPL editor* is assigned to BRIDGE participants who are interested in having a key role in the maintenance of the pattern library. As a DPL editor you are responsible for quality assurance of design patterns upon submission, and in some case there will be an iterative improvement process involving the *pattern responsible*.

The BRIDGE project is strongly founded on demonstration activities of tangible results (WP 09 Demonstration) with a subsequent evaluation and validation process (WP 10 Validation and evaluation). It is assumed that BRIDGE products associated with a GUI are demonstrated and evaluated during the course of the project life-time. The role *DPL validator* is associated with participants in the mentioned WPs, and their responsibility will be validation of BRIDGE design patterns that are under consideration. They do so by incorporating one or several design patterns into an asset (hardware, software or process) and evaluating the applicability or disadvantages of the concepts in accordance with procedures in WP9 and 10. The DPL validators may also influence prototyping and process development spanning several work packages. The process needs to be scoped and specified in collaboration with several WP representatives.

The DPL administrators have full access to modify the back-end system, and are responsible for management and maintenance of the BRIDGE DPL web site. The authors of this report are DPL administrators, also denoted as the "DPL team".

4.3 The process of submitting a pattern

Figure 5 illustrates the process of submitting a new design pattern to the DPL.

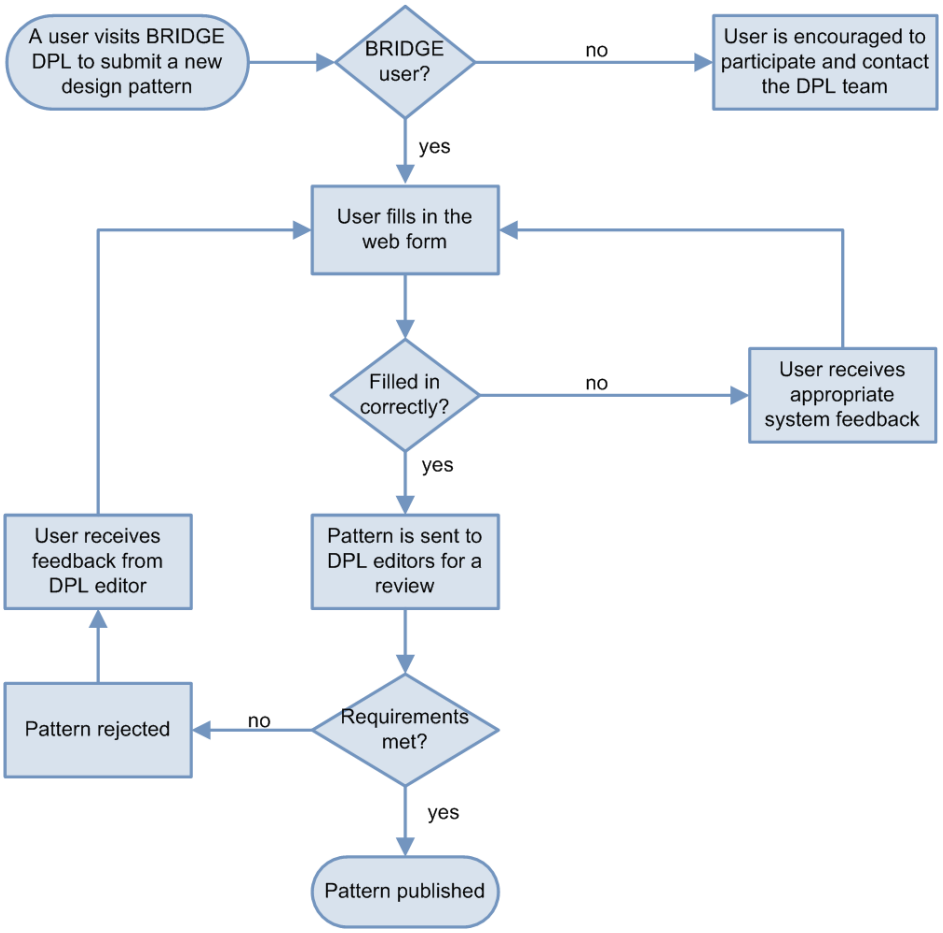


Figure 5: Submission process in BRIDGE DPL

When a pattern is newly added to the pattern collection, it undergoes a first review process checking whether all necessary fields are filled, and that the formulation of the pattern (prose language, structure-keeping) follows the conventions. If this is not the case, the author is asked for refinement. This ensures that the collection cannot just be "flooded" with patterns but that it has to pass a first quality gate in accordance with the required fields described in Table 5. Afterwards, the pattern is described and tagged in collaboration with the pattern responsible, and finally it is published on the BRIDGE DPL.

4.4 Structure and life-cycle of design patterns

Design patterns can have a variety of structures and forms - regardless of being described in books or online collections. In other words, the "pattern of patterns" is quite variable.

The patterns in the BRIDGE DPL have certain required fields or attributes, as shown in Table 5. These fields were carefully chosen as a compromise when taking into account three principles: 1. The structure of patterns should comply with standard forms of patterns in famous pattern collections and in the literature 2. The patterns should be intuitive and easy to understand for non-experts. 3. The patterns should serve the designers and developers within the scope of BRIDGE, but also be valuable in the field of crisis management more generally.

FIELDS	DESCRIPTION	TYPE
Pattern name	A short and instructive name reflecting the solution.	Text field Max 80 characters
Problem summary	The essence of the problem being addressed by the design pattern, using at most two sentences.	Text field 1-2 sentences
Pattern context / usage	This provides the context in which the design pattern is relevant, and when it should be used.	Text field (unlimited)
Solution	A description of the solution to the problem, including the background, range of use and similar.	Text field (unlimited)
Illustration	An image, sketch, or illustration	File upload
Pattern origin	The origin of the pattern , selected from list • BRIDGE • BRIDGE adapted • External	Choice field
Pattern state	Indicates the status of the pattern, and the choices are: • Just created • Under consideration (not validated) • Pattern candidate • Approved	Choice field
Pattern type	The type of pattern (only one choice allowed) • User Interface design pattern • Interaction design pattern • Application design pattern • Other	Choice field
User/role	One of the following choices: Police, Fire fighter Police, paramedic, bystander, victim, other	Choice field
Device	If applicable, indicate the device on which the pattern is relevant: Mobile phone, Tablet, Laptop, PC, surface/billboard, other	Choice field and text
Justification, references and comments	References to articles, reports, web pages, or similar	Text field

Table 5 Front-end fields of BRIDGE design patterns

The pattern name should be short and instructive, perhaps reflecting an aspect of the solution to the problem being addressed. The pattern name should not exceed 80 characters.

The field "problem summary" should be a punch line of 1-2 sentences, stating the essence of the problem being addressed by the design pattern.

The context/usage field should describe in more detail the context in which the pattern is relevant. Whenever possible, it should also provide the necessary background and information about the user, the user's task(s), the technology and more general aspects that affect the design problem.

The field "Solution description" is the heart of the pattern – and should provide a statement of how to solve the problem in the given context. The solution should always be accompanied by a sketch, diagram, illustration or picture.

To reflect the liveliness and bottom-up approach of the patterns in the BRIDGE DPL, two additional fields are added to the known pattern structure: *origin* and *state*. So far, all contributions to the BRIDGE DPL have been denoted *design patterns*. In a more strict sense however, only a qualifying subset of the submitted contributions would be designated design pattern – after being taken into use and tested over time (refer to the discussion in Chapter 3). Thus, we introduce the notion of pattern *state*. Pattern state is used to track the development of the pattern over time. The initial implementation of BRIDGE DPL has the following fields:

- "just created" for patterns that were recently submitted and needs review and evaluation
- "under consideration" means that the pattern looks promising but needs further evaluation
- "pattern candidate" states that the pattern is close to being approved
- "approved" finalizes the pattern review process and settles the pattern as a design pattern. However, this does not exclude the possibility to open the pattern for discussion in the future again due to new findings.

An existing pattern can be approved or rejected in the BRIDGE project domain. If associated with uncertainty it can be tagged as being under consideration. Adapted patterns and newly integrated ones must initially be put in the state *under consideration* (default value) and be validated as a pattern or rejected over time (cf Figure 6). A pattern idea that fails during validation might be identified as an anti-pattern. However, this should be kept as a reminder to avoid future implementations of non-trivial (surprising) design flaws.

As discussed in Section 3.2 patterns in the library can originate both from the BRIDGE project and from external sources. Three different categories can be distinguished:

1. BRIDGE: The pattern derives directly from the work within the BRIDGE project. Patterns in this category will automatically be assigned the state "under consideration". It will need to be reviewed, perhaps re-worked and finally validated through an approval process by a DPL validator (see Section 4.2).
2. BRIDGE adapted: The pattern originates from external sources, but has been adapted for use in the context of BRIDGE.
3. External: The pattern exists in other pattern collections (e.g. a standard UI pattern) an is implemented in BRIDGE products and services

Figure 6 depicts the state transition process. During a pattern's lifecycle, the different states can be changed due to new consideration or discussion. It still needs to be clarified if a pattern can be fixed disallowing any future changes. Over time, the community consisting of other readers and / or authors provide feedback to the pattern, refine and adjust formulations in shape of change requests to the original author.

Regardless of being a BRIDGE or external pattern, all patterns must be justified in a separate text entry field, see Table 5. Existing patterns must be referenced by their origin (articles,

reports, web page or similar). New or adopted patterns must be justified, and the validation event should be described here.

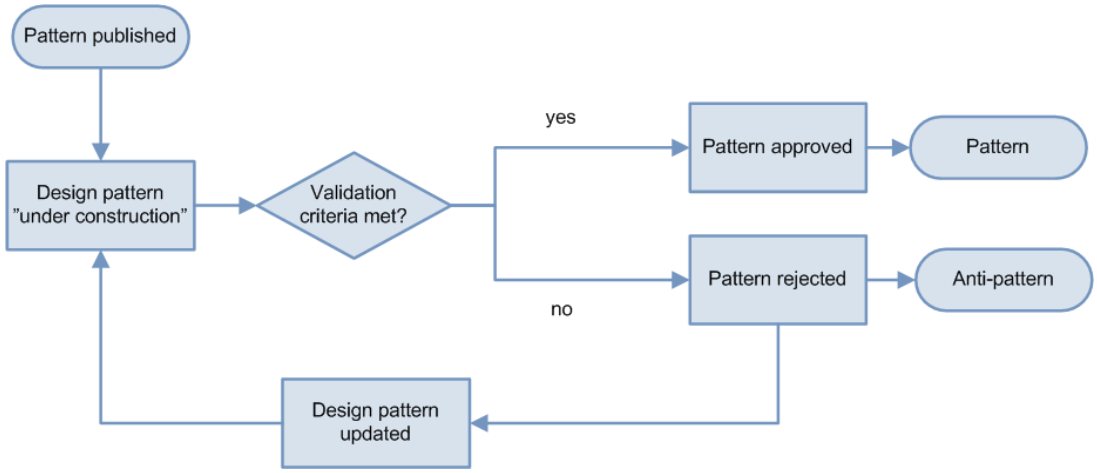


Figure 6 Simplified illustration of the main principles of the validation process

Pattern type should also be specified:

- **User Interface design pattern** - if the pattern describes the graphical representation of elements in a user interface
- **Interaction design pattern** - if the pattern describes interactions with elements of an user interface
- **Application design pattern** - if the pattern describes the overall concept of an application type
- **Other** - if the pattern does not fall into one of the other categories

Whenever relevant, the user or role associated with the design pattern should be specified, and also the type of device the pattern is adapted to.

4.5 Future extensions of the BRIDGE DPL

Depending on the process and progress of the BRIDGE DPL, the following features are considered in future updates:

- Differentiate or extend the context field, and integrate tags for device, user, and task into the description of a pattern
- Integrate an anti-pattern attribute to mark a surprisingly failing proposal of a solution
- Ranking of patterns - based on the feedback from readers, and (un) successful implementations or other kinds of feedback
- Contributions from BRIDGE external (possibility for direct submission to the DPL)
- Dependencies among patterns and abstraction to a pattern language

5 Walk-through of the BRIDGE Design Pattern Library

This chapter presents the BRIDGE Design Pattern Library from the user experience perspective, with screenshots exemplifying main features and functionality. The BRIDGE DPL is intended to handle the complete pattern creation and evolvement process throughout the project's lifetime. The following sections describe the front-end and back-end functionality of the BRIDGE DPL.

5.1 Front page

When accessing the site at http:// http://pattern-library.bridgeproject.eu or via the BRIDGE project homepage, a welcoming page is shown, providing a short statement about the site's intention, as well as links to information and the Mendeley literature repository (cf. Figure 7).

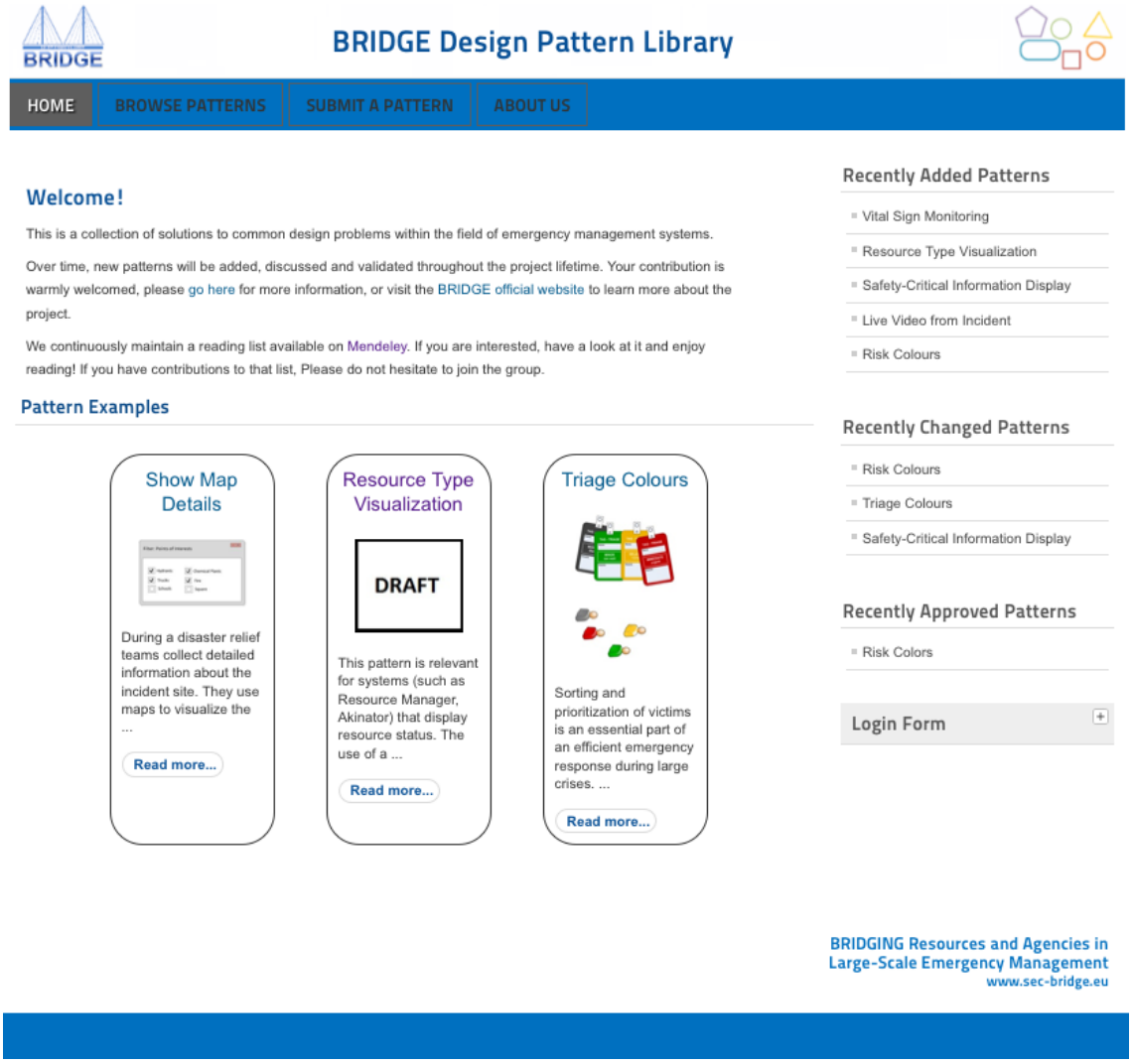


Figure 7 The BRIDGE DPL front page

Below the welcoming section, a dynamic choice of available patterns is presented as a preview. The right column is used for placing dynamic overviews of recently added patterns and recently changed patterns, respectively. Recently approved patterns are also shown. This way, we want to show that the site is actually changing and to inform the visitor what the latest actions are.

Below the links to the latest pattern changes, the login form is displayed, allowing the user to log in or create a new account. This menu is foldable to gain space.

The available options for the front-end are given in the navigation pane at the top; from here, the available patterns can be browsed, a new pattern can be submitted and additional information about the pattern library and the DPL team can be accessed.

5.2 Browsing the Design Pattern Library

When accessing the "Browse Patterns" tab, the currently available patterns are displayed as smaller preview cells in rows of three items (cf. Figure 8). The cells provide the pattern name, the solution illustration and the context in which the pattern is to be applied. This preview mode serves as a teaser to give the visitor the chance to see whether the pattern fits into her problem context. If the reader chooses a certain pattern, she is taken to the *detailed view* where the detailed problem description together with the solution and further information about the pattern is shown[6] (cf. Figure 9).

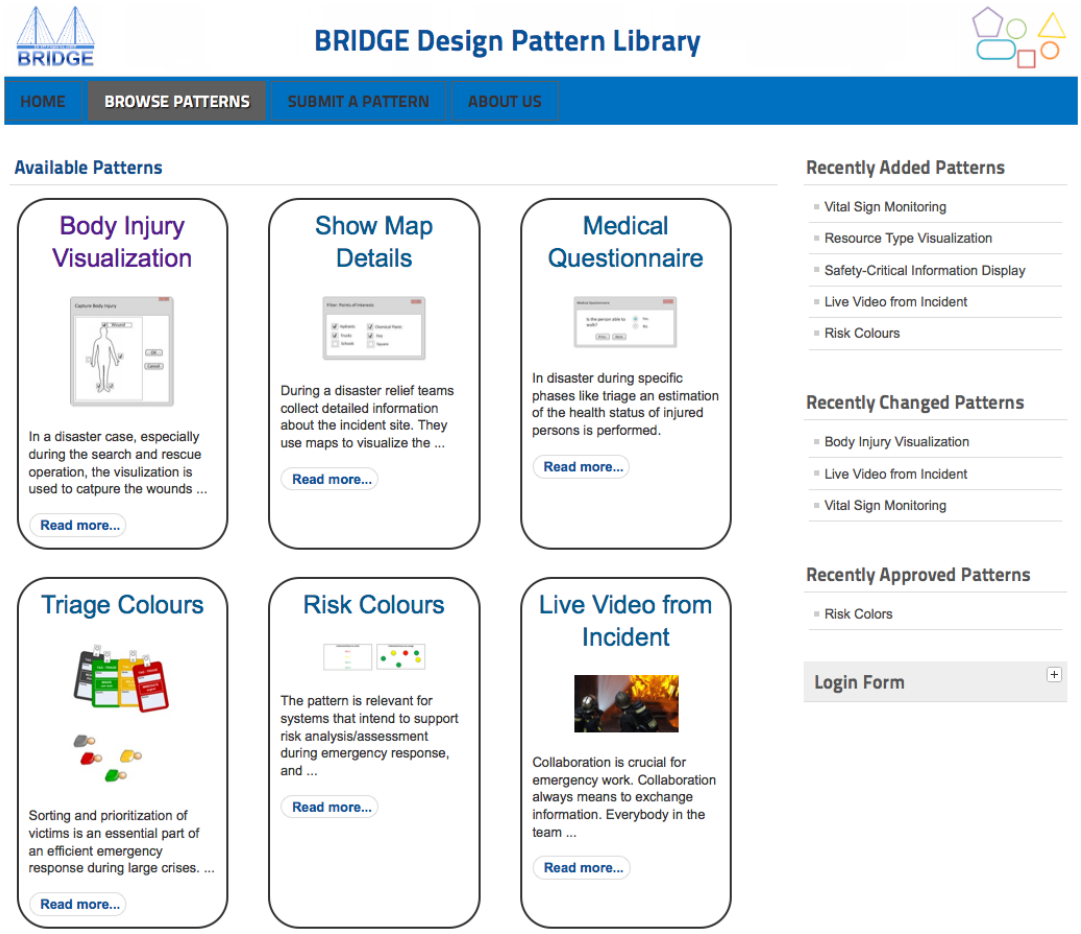


Figure 8 Browsing the patterns in the library. Each pattern is shown with its name, a small version of the solution illustration and a teaser text taken from the pattern's context

The pattern details view provides all the fields that are currently defined for a pattern (cf. Figure 9). Starting with the pattern *name*, the *context* in which it can be applied followed by the solution. Thereafter, a scheme, picture or image *illustrating the solution* is shown. After these canonical fields that are common in most pattern languages, the pattern's *origin* is shown. This marks whether the pattern can be found in other context and directly reused within the scope of the BRIDGE project (*"external"*) or needs to be adapted to suite the problem domain's needs

[6] In later iterations, we will add multiple views that cluster patterns by type, role or device in order to facilitate navigating a growing number of patterns.

(*"adapted to BRIDGE"*). A third alternative is represented by the origin field *"BRIDGE"* meaning that the pattern or pattern idea, respectively, is newly derived and needs evaluation to reach an approved state or eventually is rejected in the future development according to evaluations of prototypes.

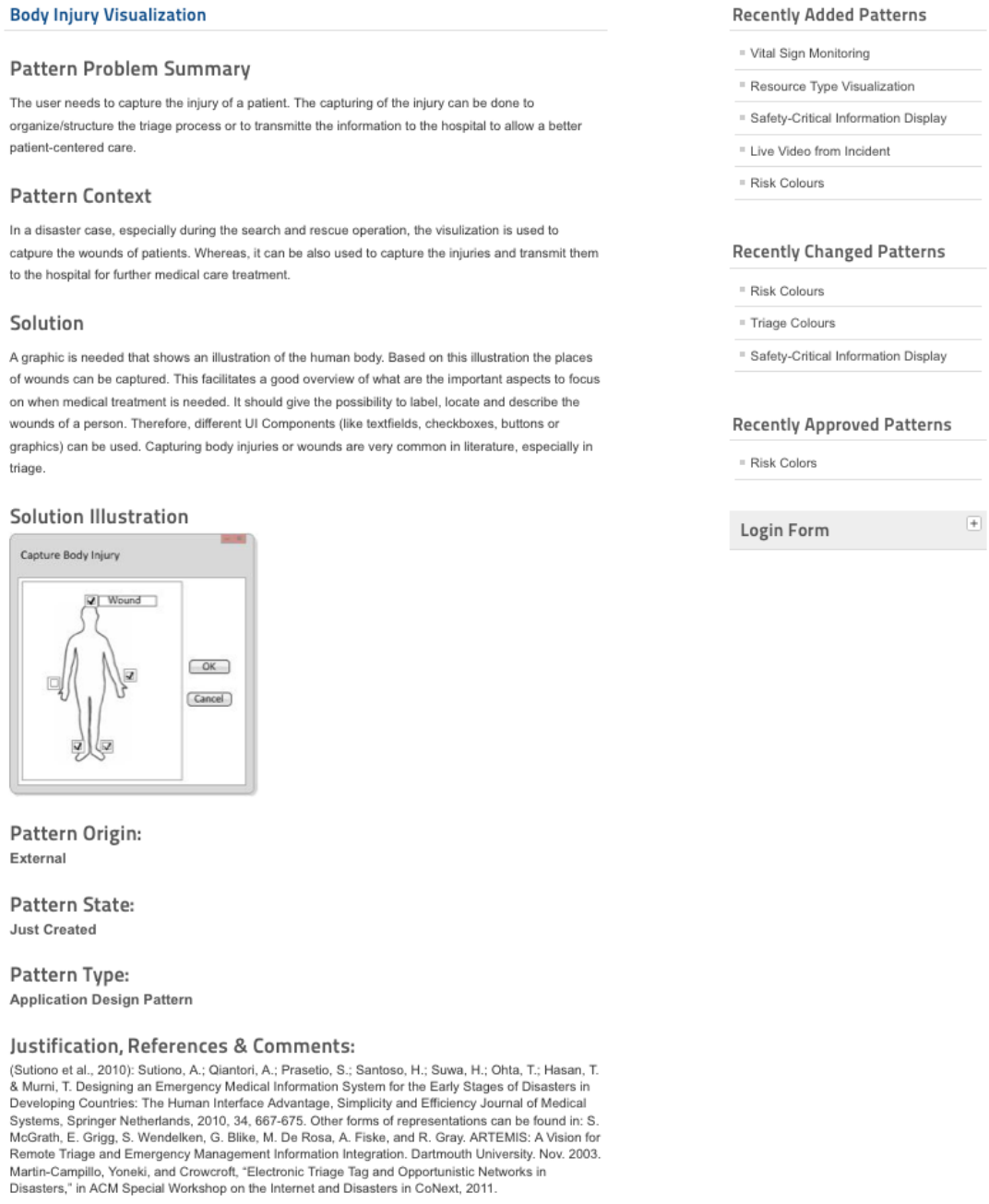


Figure 9 The pattern details view showing all fields belonging to a pattern

The *pattern states* reflect the review status of a pattern, for details refer to Section 4.4. The *pattern type* refers to the pattern classification system (UI pattern, interaction pattern or application pattern) as described in Section 2.2.

Every pattern can also be connected to a *user role* and a *device*. These fields are optional and can be extended on demand by the site's administrators. Roles and devices can be very specific, (e.g., incident commander, doctor, mobile phone, tablet, PC, etc.) or represent user groups or

device classes (e.g., medics, assistance, handheld device, wearable device, etc.). The first set of roles and classes is chosen based on the submitted patterns.

The final field belonging to the pattern directly is called *Justification, References & Comments*. Here, references, links, documents or videos proving that the pattern was already applied successfully are listed. The more references are collected over time the stronger that pattern's reputation becomes.

Besides the pattern fields, metadata like the creation date and initial author of the pattern is shown. This way, readers can also directly ask the author for clarification or contact him / her for feedback. From the creation date, readers can also derive for how long the pattern already existed in the pattern library.

Looking at the dynamic quick links on the right hand side, it can be seen that the pattern currently being read and listed in bold face to indicate the connection between the currently displayed pattern and the presented choice.

5.3 Browse patterns by filters

During the development process and based on the submissions so far we came up with the idea to present additional browsing views. The current version shows all patterns available, sorted by their submission date.

However, we expect that the amount of patterns will grow in the future and therefore we are thinking of views that sort pattern based on user roles, devices, types or states. Appendix 3 (second page) shows a prototype of a possible view layout. This work is currently under preparation and will be incorporated into the site as soon as the number of patterns exceeds the number of 20. In our opinion, this represents a threshold at which clustering makes sense.

5.4 Submitting a new pattern

Before publishing the DPL online, we asked the project members to fill in documents with pattern suggestions as shown in Appendix 2. At the time of publishing the site, we also provided an online form which registered users could fill in and thus submit a pattern to the review process described in Section 4.3.

Error! Reference source not found. Figure 10 shows the submission form with the fields described in the preceding section. Fields marked by an asterisk are mandatory. A script checks whether they are filled before and eventually marks missing fields red before the submission in completed.

The submitted pattern is then stored in the DPL database and available in the back-end. There, responsible perform the pattern quality check and publish the pattern or ask the author for refinement of the pattern if needed.

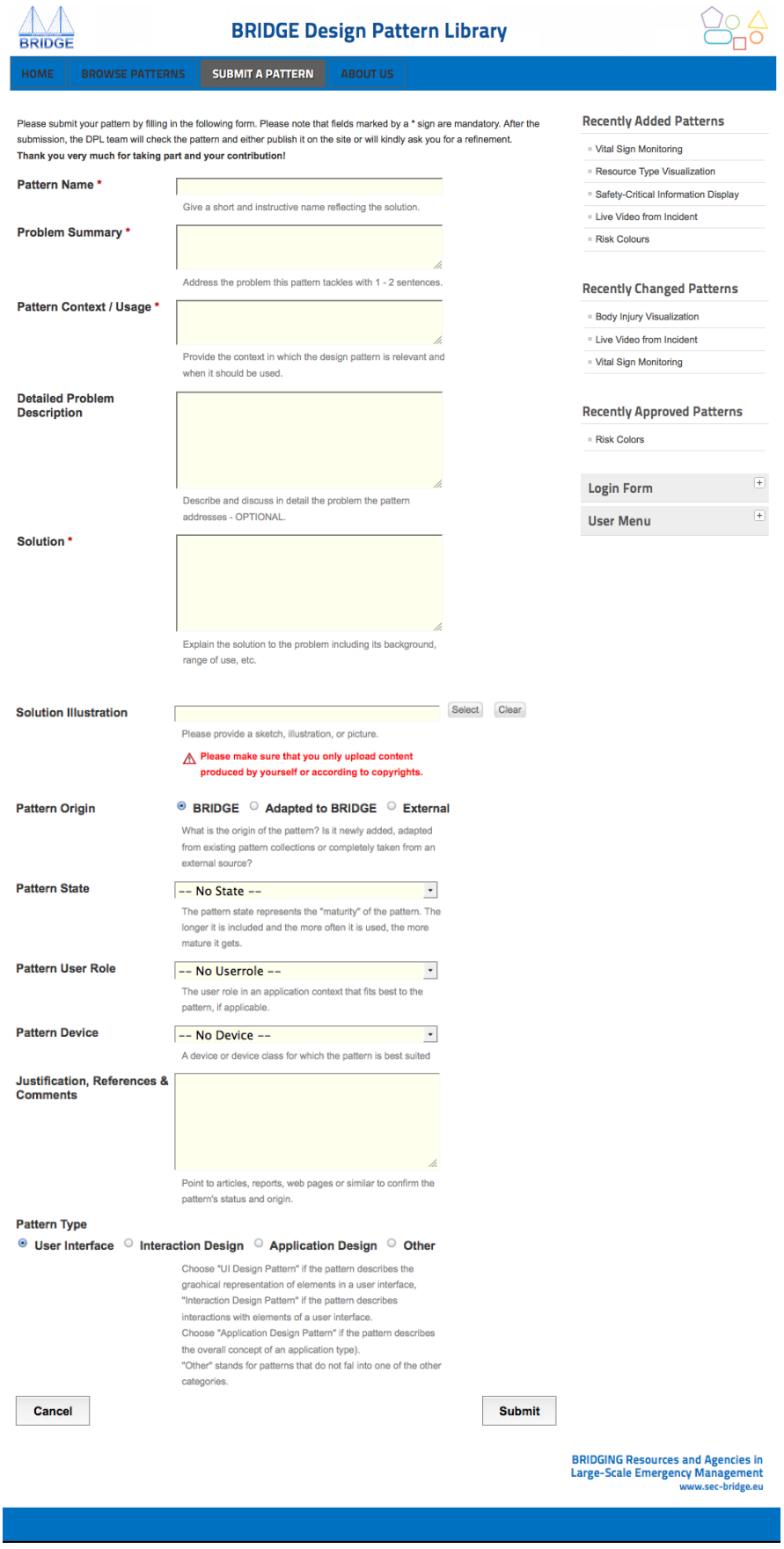


Figure 10 The pattern submission form.

5.5 Item Management in the back-end

As mentioned in the sections before, the Design Pattern Library has two components, the front-end for site visitors and the back-end for publishers with responsibility for the pattern quality check and site administrators. From the back-end, which is available by appending *"/administrator"* to the website's address, the appearance of the website can be modified together with menus and displayed components, like the pattern quick links shown on the right hand site. For articles and other custom items, a powerful sorting and managing functionality is available originating from Joomla!'s initial intention of being a Content Management System.

The DPL library is a custom component that builds on top of the Joomla! framework. Therefore, the listing and administration functionalities of pattern are very similar to the ones of articles. When accessing the *Pattern Management* component in the back-end, users can choose between listing the available patterns, devices, user roles or pattern states.

Each list provides general information about each listed item. Most important is the *status* column of each item. A red circle indicates that the items exists but is not available at the front-end, a green checkbox means that the item is published an available.

Additionally, items can be archived or trashed. In contrast to trashing items, archiving an items is meant for preserving it without putting it online or changing it anymore in the back-end.

Figure 11 shows a screenshot of the list of available patterns. Submissions from the front-end are listed here directly, automatically being assigned the *"not published"* state. The views for user roles, devices and pattern states are similar to the one shown.

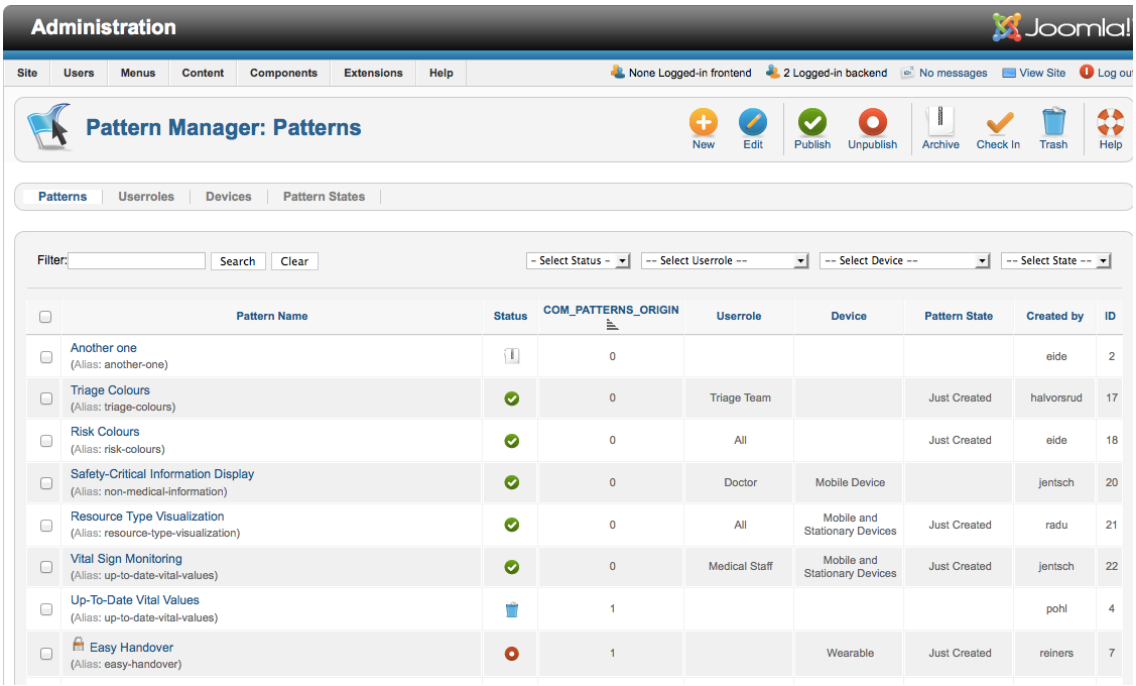


Figure 11 Pattern Management in the back-end: The status of each listed item can be changed directly by clicking on the icons.

By clicking one of the items, a detail view is shown with the already available field data. Additional metadata like the last modification date or schedules for automatically (un)publishing the item by the system is available in this view. A screenshot is available in Figure 12.

Following the example of a pattern, new devices, roles or pattern states can be created, modified, archived or trashed. The publishing states can be set accordingly.

The back-end provides the tools necessary for maintaining the DPL and updating the fields without the need of touching the source code since all data is kept and updated in a database which is accessed and modified automatically by changing data in the forms.

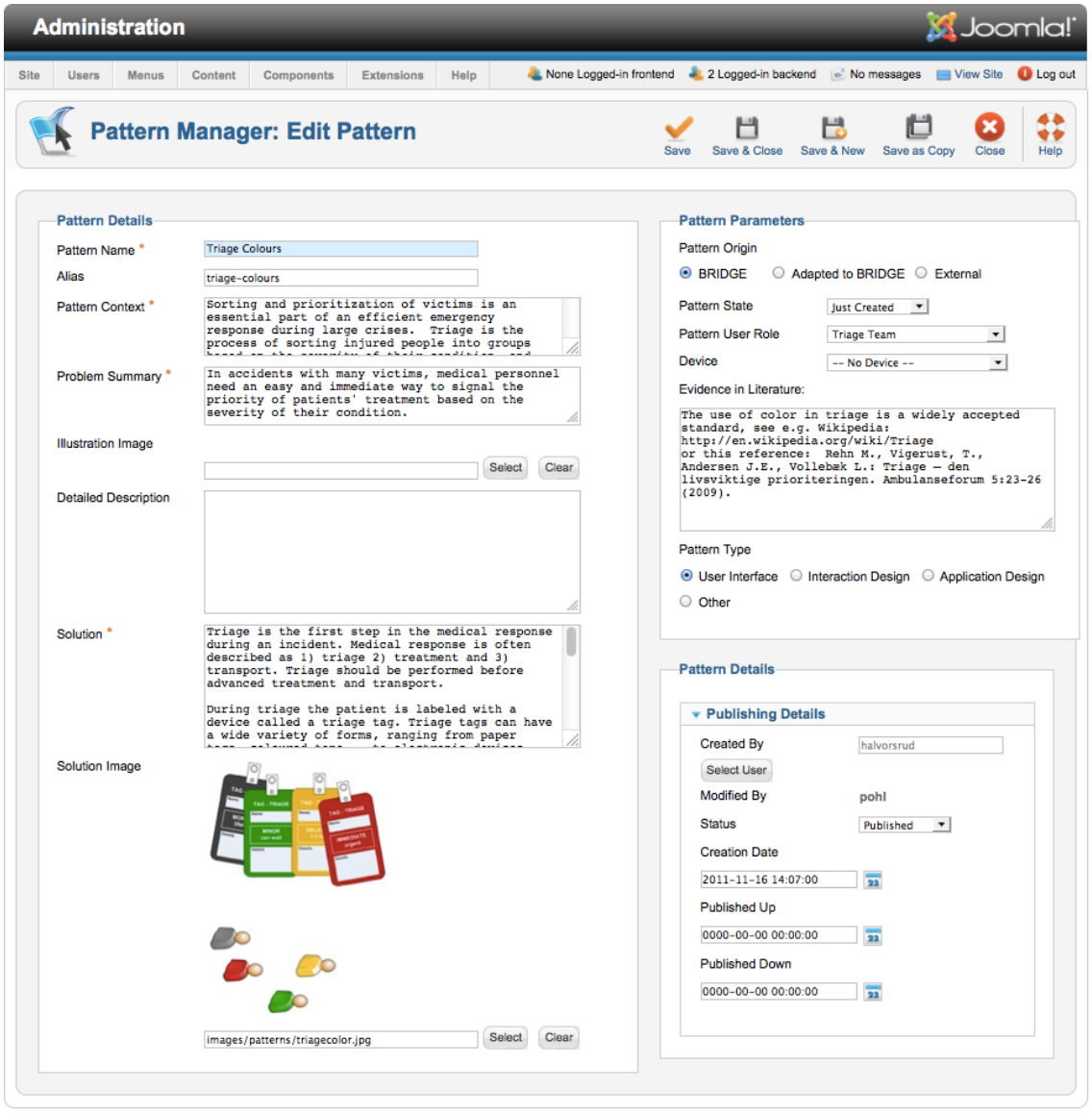


Figure 12 Detail view of a pattern in the back-end.

5.6 Outlook: Discussion, proposals and refinements

With this deliverable the features described are available concentrating on showing available patterns in a browsing and detailed view. Another important feature is the possibility to directly submit a pattern to the review process.

However, the project goes on; by providing the platform, our task now is to keep on encouraging domain analysts, application and interface designers, respectively, to continuously contribute their findings to the Design Pattern Library throughout the project's lifetime. Thus, the DPL is not closed with the submission of this deliverable but has just started its evolution.

When talking about evolvement, features to comment on existing patterns, keeping track of changes and provide ways to edit or extend existing patterns via the front-end and thus making

the DPL a community portal for openly submitting, refining and discussing findings that were made during the project's lifetime.

We will see what ways are appropriate and easy to follow concerning commenting, eventually voting or preparing pattern ideas to achieve the next state on its way towards an approved pattern. Eventually, new ways for clustering and browsing patterns will need to be found.

6 Principles and design patterns in emergency management

A literature review within the field of emergency management systems was conducted during the initial work on the DPL. The purpose of the review was to establish the baseline within this domain, and to identify potential design principles and practices that can make relevant contributions to the library. We begin this chapter with an introduction to the most general design guidelines and principles within this domain.

6.1 General guidelines and principles

Emergency management involves a great variety of different personnel, ranging from tactic personnel working in the field, to operative and strategic personnel working from emergency centrals. All the individuals involved have different requirements and needs according to their role, tasks and the equipment they use. First responders need mobile equipment that supports hands-free communication, as their hands are busy with other tasks, such as driving to the incident site. Operative personnel in an emergency central, however, are better off with larger displays, enabling them to get a complete overview of the situation. Thus, for emergency response systems specific characteristics must be considered, both for the mobile devices interaction in the field, and the more fixed interaction taking place in the control post.

Carver and Turoff (2007) emphasize the following aspects for implementing a successful emergency management system:

- Exchange of information including information from in the field: Exchange of information is crucially necessary in order to obtain/maintain a common situational awareness across the involved personnel.
- Just-in-time decision support: All information must be summarized to save time and to enable efficient decision-making.
- Focus user's attention on most important facts without the feeling of interruption: Emergency personnel that are occupied with carrying out orders and/or saving lives should only be interrupted when this is strictly necessary. Place for creativity to find a solution by the user: Due to the dynamic and complex nature of emergency situations, it is important that the systems enable its users to improvise when solving a given problem.
- Trust building between team members: Response teams working in large-scale emergency situations are typically assembled ad-hoc. As team-members may not have worked together in the past, trust-building becomes essential.
- Facilitate workflow dependent communication; gain additional information over time: For certain emergency situations predefined workflows exist that should be adopted by the emergency response personnel. A system should therefore facilitate workflow-dependent situations.

Given the critical nature of emergency situations and the variety of roles, tasks and contexts involved, it is crucial that the end users themselves play an active role in the design process. This ensures that their requirements and needs are fulfilled in the resulting system (Carver & Turoff, 2007). The users should also be strongly involved in evaluation of the system. In particular, it is important to make sure that the user interface does not disrupt the user in performing tasks, but rather support him/her in gaining sufficient awareness of the situation. Therefore, the system should mediate crucial information and feedback, taking work practice and overall workflow into account. Flentge et al. (2008) emphasize the following aspects in the design of user interfaces:

- *Reduce the complexity:* The UI should reduce complexity and facilitate an overview of the situation and simplify communication of information
- *Focus on tasks and devices in the UI design:* Different roles and tasks needs specialized equipment. Relief teams should be supported through devices they know well and suited for working in the field (e.g. not a laptop for a firefighter in the field)
- *Operable by novice and experts:* The system should be designed for the usage by both professionals and novices
- *Flexible UI for different situation (environment and device):* The full range of the application area is not assessable from the very beginning, so flexibility in UI design is needed
- *Security and privacy:* Sensible data must be protected for unauthorized access, and decision making must be traceable

Although these principles are applicable for many types of systems/applications, they are particularly important and sensible for emergency management. Privacy and security of patients for example, is a very sensitive and special topic. The system/application should be useable from the very beginning, both by novices and other involved parties that have not undergone detailed training. Ignoring the aspects above in the area of emergency management systems may have more severe consequences than it would have in other applications and areas.

In summary, it can be said that the design of information systems for emergency management needs special design activities due to specific characteristics of a crisis situation.

Turoff et al. (2004) presents a framework for the design and development of emergency management systems, based on an extensive literature review. They present the following design principles for emergency response systems:

1. *System Directory:* Hierarchical structure for all the data to browse information with a text search and to group related information.
2. *Information Source and Timelines:* Data must be specified by its source, time, state, location and links to information already present within the system to reach the maximal overview.
3. *Open Multi-Directional Communication*: Communication structure open for all interacting parties.
4. *Content as Address*: User specifies events as important for their tasks; content-based deliverables.
5. *Up-to-date* information and data displayed
6. *Link of data objects*: Information which is interrelated must be linked (semantically)
7. *Authority, Responsibility, and Accountability*: Information must be visible to the right persons, which have the authority and the overview of what is allowed and what is not allowed (or what is conflicting and what is not conflicting)
8. *Psychological and sociological factors*: Encourage and support the psychological and social needs of the crisis response team to build trust and reduce stress.

Furthermore, it is pointed out that *information overload* is one of the most challenging problems within crisis responses. Therefore, a user should be able to adapt the system to support his/hers cognitive processes.

6.2 Application design

Most systems in emergency management are based on an easy-to-use GUI with special attention to the needs of the user. A variety of topics and modules exists, ranging from knowledge representation, maintenance of situation awareness, to fully developed application platforms.

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. Hanratty et al. introduces a knowledge visualization approach that enables visualization of the agent decision space, where agents represent human beings and software agents. The representation shows the past situation and the current evolving situation. Through the visualization the user gets an insight into the complex situation, supporting the decision making process.

Nestler & Klinker (2007) describe a mobile user-interface for disaster management and health care personnel performing triage[7]. During the emergency management, patients are assigned to more than one relief team and vice versa. Hence, three different representations are needed to give an overview for all relief units:

- Static knowledge (treatment, coordination procedure etc.)
- Patient related information (particular history etc.)
- Peripheral information (number, states and position of patients and relief workers etc.)

The patient condition is evaluated by the relief units using questionnaires, and the patients are classified according to different care categories. The state of the patient is captured and stored for other relief units by addressing a number of questions such as "*Able to walk?"* (Figure 13, left).

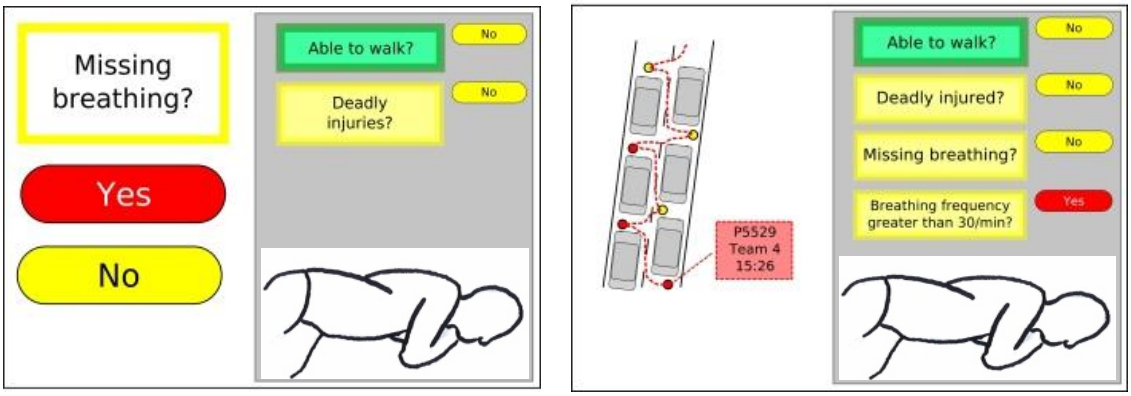


Figure 13 Left: Questionnaire for classifying the current patient. Right: Patients on the incident site and the path followed by the search & rescue, and information of a specific patient (adopted from Nestler & Klinker, 2007)

To get an overview of the patients within an area, Nestler and Klinker make use of a map with standardized symbols for marking the position of patients, their state and number, and also information about relief units (see Figure 13, right). Detailed information about a patient can be requested from the map, where information about the patient's condition is included (such as the first contact of the relief unit, including the name, position and time). By switching to another map, one can display the distribution of all patients in a certain area (see Figure 13, right).

[7] Triage: A process for sorting injured people into groups based on their need for or likely benefit from immediate medical treatment (from the free dictionary at http://www.thefreedictionary.com)

Besides the standard UI interaction through a visible interface there are also other techniques that can be applied in emergency management. Bader, Meissner, & Tscherney (2008) make use of "smart rooms" at the control post. They propose a technique that allows users to interact with the system without the use of a visible interface. A room equipped with sensors, cameras, and actuators allows the system to learn actions by monitoring gestures, movement and speech. For such an application the following guidelines were developed:

- Support interpersonal communication, easy to handle different communication channels
- Inter and Intra-group communication and interaction
- Define information for users based on their roles, and deliver the right information
- Interaction and communication must be recordable (usable in court)

Another work of interest is that of Sutiono, et al. (2010) who describe a medical information system for disasters in developing countries. Information about patients is captured in a specific user interface illustrated in Figure 14 and Figure 15. The first tab supports the categorization of the patient based on condition and state. The second tab is a checklist for the wounds identified. This supports the medical staff in handling patients based on their conditions.

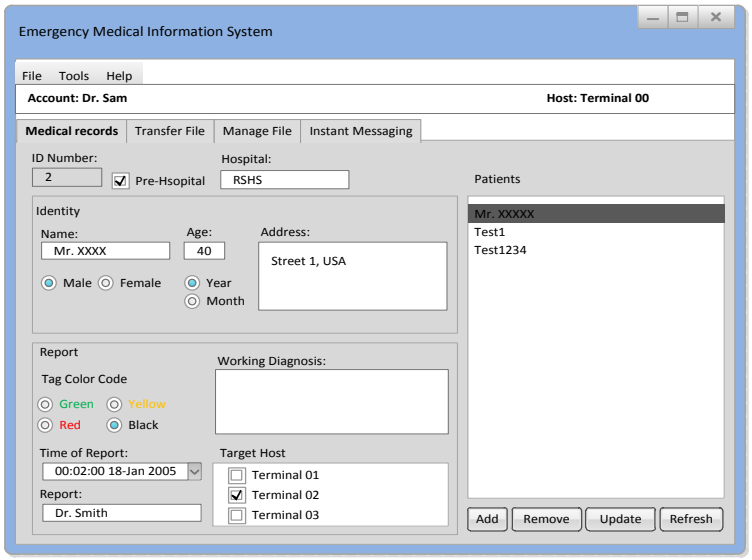


Figure 14 EMIC user interface for the Medical Record (adopted from Sutiono, et al., 2010)

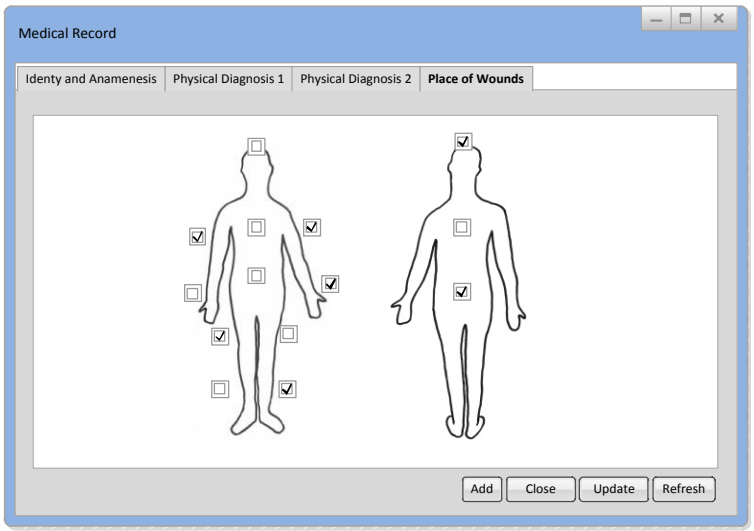


Figure 15 EMIC user interface for the Place of Wounds (adopted from Sutiono, et al., 2010)

The research effort of Monares et al. (2011) describes 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 different functionalities (see Figure 16, bottom-center)

- Navigation: Automatic navigation through the displayed map is achieved by tracking the location of the fireman using a GPS. On-demand navigation is performed through clicking on the device.
- Destination: A destination point can be chosen (address or coordinates) and then the system navigates to this point.
- Distance: Calculates the distance between two points.
- Information: Information management is separated in different parts (see Figure 16):
    - Information about trucks, like state.
    - A list of ongoing emergencies, facilitating detailed descriptions of each entity
    - Points of interest are selectively displayed in the map (vehicle, hospital, hydrant etc.), along with details about selected points of interest
    - A mailbox functionality for the exchange of information with others or the command stand (pictures or any other files). The information is exchanged via GSM to the command stand and distributed to others in the field. If the command stand is not available, Wi-Fi is used to form an ad-hoc network to exchange information between firemen.
- Current Location: Shows the current location of the user of this device.
- Fire truck: Shows the position of all fire trucks in the field. Additional information about fire trucks is also available as each truck is equipped with a mobile device transmitting the current location and the most important facts about the truck (capacity of water tanks, and information about other equipment).

The system is realized in a client-server architecture where the server acts as information repository. In addition, all information communicated via the radio channel are recorded and delivered in real-time to all firemen in field. This ensures that no information is missed. The information supports each firefighter in case of improvisation. All information, including the communication via the radio channel, is logged and stored for later analysis and improvement.

The MobileMap system may help to derivate patterns for the same or similar problems, like the selection of points-of-interests or the layout with different tabbed panes. One problem of this system however, is that information sent to the control post is not displayed in the map, nor do they have any metadata (GPS) captured, like for collected pictures.

The EU-funded project COPE[8], describes several GUIs that where introduced during their work. A very valuable source here is the public deliverable D2.3 (Norros, Liinasuo, Savioja, & Aaltonen, 2010). Fire service was the focus in the COPE project. The roles that a recognized are fire fighters, sector commander and incident commander in the field. Therefore, a fire fighter system called FRS-HW (First Responder System – Human Wearable) was introduced that consisted of helmets carrying devices (e.g. cameras and wearable sensors). The information

[8] http://cope.vtt.fi/

from the devices can be transmitted to the section commander to gain an overview of the situation. The work to be done is seperated into tasks that are shared with the responsible sector commanders. Both the sector- and incident commanders use a map display to gain location depent information (e.g. of specific points-of-interests, location of team member, etc.). Figure 17 illustrates the system used for gaining an overview via a map.

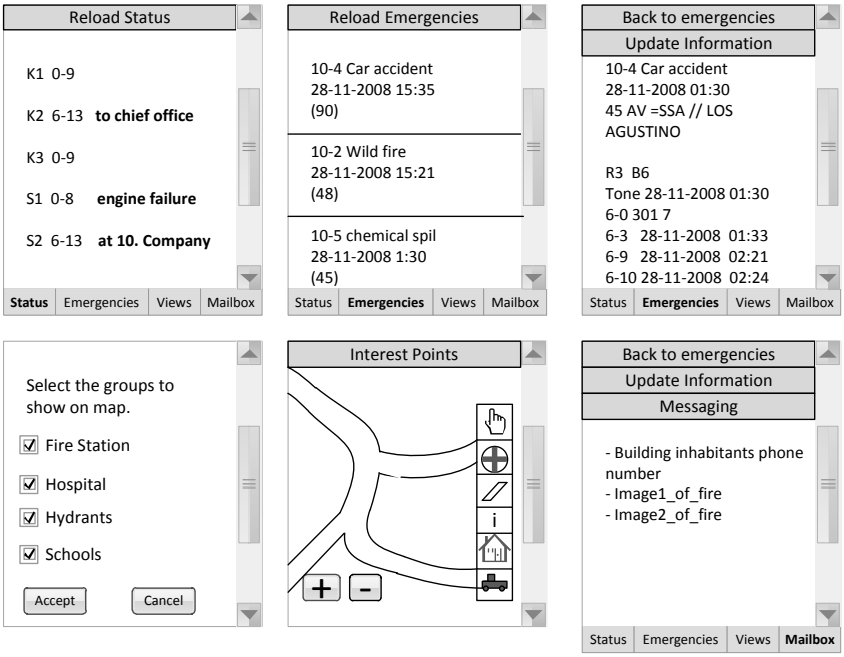


Figure 16 UI of the MobileMap Application (adopted from Monares, Pino, Herskovic, Rodriguez-Covili, & Neyem, 2011)

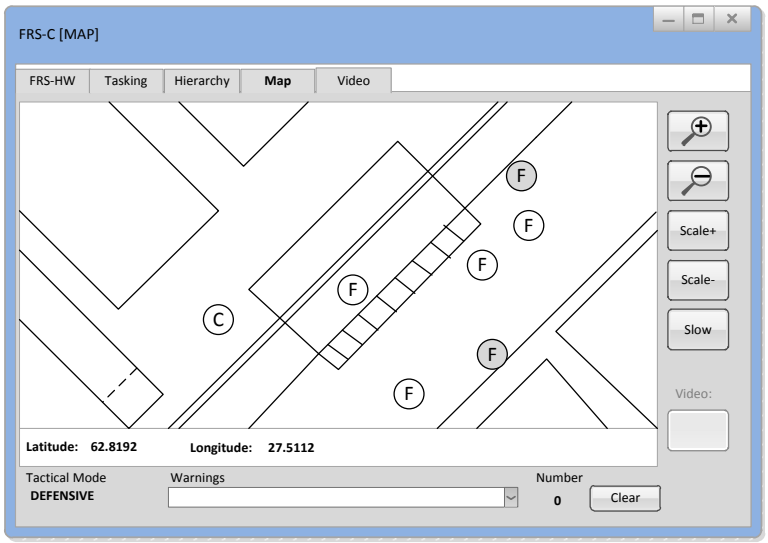


Figure 17 Map view of the system for the Sector Commander. F describes the position of fire fighters and C the position of cars. (adopted from Norros, Liinasuo, Savioja, & Aaltonen, 2010)

Based on the design practices above, we have derived the following design patterns: Body-Injury-Visualization, Show-Map-Details and Medical-Questionnaire. The patterns are included in the design pattern library and additional sources for similar design solutions are included as evidence within these patterns.

6.3 User-interface design

While the previous section focused on general application designs for emergency management systems, this section focus on the more specific parts of UIs for emergency management.

Displaying all information – first of all on small devices – can be a very challenging task. Symbols and signs can be used to overcome these challenges, and to avoid overwhelming the user with information. The EU-funded project Indigo[9] describes such a symbolic representation. Newlove-Eriksson & Hermansson (2010) provides a detailed overview on their work in this area, and an 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. The work of Fitrianie and Rothkrantz (2007) provides an example of an *icon language* (see Section 6.4).

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. From this study a pattern catalog was extracted, resulting in 19 design patterns grouped into 4 sections (Acuna, Diaz, & Aedo, 2010):

1. Emergency Response and Recovery (Awareness for First Responders, Awareness for Command Control Operators).
2. Planning and Mitigation (Decision Support System for Planning, Collective Memory / Data Archives)
3. Information Management (Information Presentation, Map-based Navigation, Search, Sort and Filter Information, Data Authoring, Up-to-date Data, Contextual and Related Data, Extended Data, Alert Notification, Fast-Loading Pages, Contextual Help, Direct Manipulation)
4. Access For all (Style Sheets for Multiple Media Types, Pages Accessibility, Printable Pages, Hand-held Devices)

A short description of the design patterns can be found in (Acuna et. al. 2010). A sketch of the design pattern called “Alert notification” is provided in Table 6. Details about the patterns are not found on the Internet, but a detailed description of two of the patterns can be found in (Montells et. al. 2007), along with another example of this work called WEMS Directory (web-based Emergency Response Systems).
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. Within the work of Carver & Turoff (2007), they view disaster managers as a kind of game. They base this metaphor on the notion that gamers, just like emergency personnel, must have a full overview about what is going on. Therefore, best-practice from playing or designing video games can be a valuable source. In addition, there are also standard design patterns in the context of multimedia which may be used in the traditional design process to find generally valid solutions. The multimedia focused patterns are relevant for displaying images, sets of images, videos and similar. Such patterns can also be transferred to the design

[9] http://indigo.diginext.fr/EN/index.html

processes for mobile devices. The following examples from the Welie[10] Pattern library correspond to multimedia:

- Minesweeping: With mouse-over, additional information to graphical elements can be shown
- Scrolling Menu: Show items on a scrolling menu for selecting an image from a list
- Guided tour: Teach how to do something
- Slideshow: To view images and photos (with interaction buttons, like previous and next button)
- Stepping: For viewing linear ordered items
- Carousel: Show items in circular fashion
- Overview by Detail: Show an overview of items (per pictures), after the selection the detail information is represented
- Thumbnail: Display small version of multimedia
- Virtual Product Display: Interaction with a virtual product

The following additional design patterns from the UI[11] Design Pattern Library can be added:

- Image Zoom: Zooming within images per mouse click, touch etc.

Additional examples especially for design pattern of mobile devices can be found within the Flaminco[12] design pattern library of the BRIDGE participant Erik G. Nilsson (Nilsson, 2009).

DESIGN PATTERN	
Name:	Alert Notifications
Background/ Context:	During emergency response and rehabilitation, it can be requested to present users with notifications or alerts containing possible important information. […]
Emergency phase/role:	Emergency Operation – Rehabilitations/Recovery Command Control Operator
Problem:	During the emergency response or rehabilitation, notifications about a particular situation may be needed to present the users in case the data is important or related to the current event. […]
Solution:	Notification should be displayed to users in an autonomic way, presenting a small and descriptive summary […]
Related patterns:	Notification can be implemented with **floating windows**, […]

Table 6 Alert Notification as a design pattern example (adopted from Acuna et. al. 2010)

Another approach is the so called *fisheye view*, where a specific part of the system is shown in detail and the rest only as a sketch – which is a specialization of the "overview by detail" pattern. The fisheye is like a lens on a specific part of a map or similar representations. In addition, developing applications for smartphones show also standards on how to handle different kinds of applications. Within this UIs[13] Pattern Library examples for the pattern "Photo Thumbs" can be found, which moves the previously described pattern "Thumbnail" for standard devices to the application for mobile devices.

[10] http://www.welie.com/patterns/index.php
[11] http://ui-patterns.com/patterns
[12] http://www.flaminco.net/
[13] http://pttrns.com/photos

Another interesting finding is that images showing disaster situations do not necessarily suffer from low resolution (Alers, Gunawan, Brinkman, & Heynderickx, 2010). As long as the most important things are visible on images, it is no problem to reconstruct the disaster situation from a low quality picture. This gives the opportunity to scale down the image quality due to network constraints, without causing a drastic lack of information about the disaster. High quality is also unnoticeable by viewers during disaster management (Alers, Gunawan, Brinkman, & Heynderickx, 2010).

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 according to the content it holds. An example of this is given by the Task-Handler of the WORKPAD[14] project, which is a user interface designed to support the overview of personnel, teams and their corresponding tasks. Figure 18 gives an overview of the Task-Handler interface.

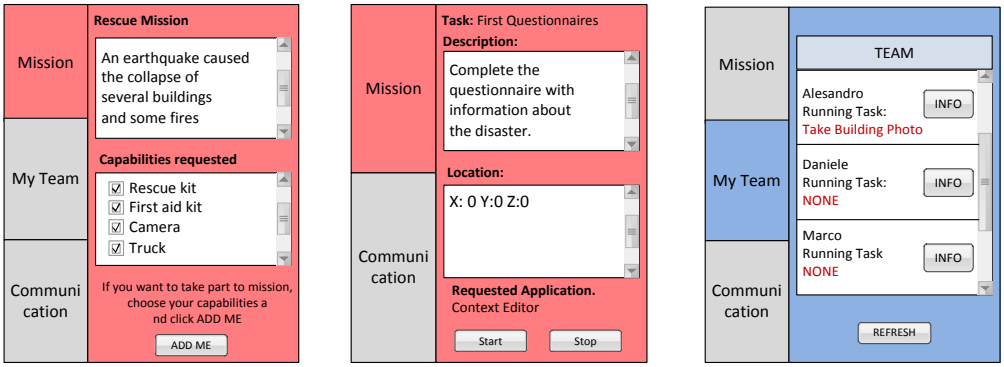


Figure 18 Task information from team leader view (left), Task for a team member (middle) and Status of all team members (right) (adopted from Humayoun, et al., 2009)

The WORKPAD application may be a potential source for derivation of patterns for the emergency area, especially for the task assignment or task handling. General patterns, such as slideshow and thumbnail, are applicable in many different systems. Still however, before applying them within the emergency management context, one must first assure that the pattern actually works as intended within the given system.

6.4 Interaction design

Traditional human-computer-interaction (HCI) design solutions can be cumbersome in the area of disaster management. Interacting by means of mouse and keyboard can be a very complex and laborious task during stressful situations. Therefore, new approaches must be created that simplify the interaction of the command leader, relief units, etc. with the system. These interaction models must support the user and help him achieve a natural interaction with the device.

The use of multi-touch interaction that enables multiple users to interact with a system simultaneously has been suggested for use in emergency management (Flentge et. al., 2008). The work of Micire & Yanco (2007), describes such a multi-touch panel, which allows 4 users to work simultaneously, updating a map of a disaster area. Traditionally, such maps are updated using pen and paper in a sequential manner by different relief units. Using the multi-touch panel as described above, up to 4 people can work in parallel to update the map according to the current state of the situation.

[14] http://www.workpad-project.eu/

Emergency units often focus on physical stimuli and not on things coming from a software application (Humayoun, et al., 2009). To overcome this issue, the attention should be captured by the software system. To prevent the user from interruption in his/her work, this should be done only when necessary (e.g. when new and important information is available). This can be achieved by the use of popups and sonorous alarms (Humayoun, et al., 2009).

Another approach is described in the work of Krahnstoever et. al. (2002) who introduce a multimodal interaction method. This method combines gesture and speech recognition methods for interaction with the user interface. Therefore, real-time processing for gesture and speech recognition is necessary. In addition, those two interaction methods must be fused to get the best interaction result in real-time. Simple commands in combination with gestures tell the system what must be done. For example, "Zoom in" with the focus on one specific point or area instructs the system to switch to a detailed view of the pointed area. The system contains a high resolution screen with a camera that tracks the movements/gestures and a microphone that records the voice. It facilitates the interaction of a person with a map showing the incident site, for example it helps to zoom in to the map or to navigate through the whole map. The map also contains some points-of-interests (like the location of hospitals etc.).

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. This is achieved by offering a personalized view which can be used by a single-person or a group of people working together. The personalized view is offered via so called Fovea-Tablets that can be put on top of the tabletop. On the table they act as a lens, showing personalized information relevant for that person/group, or detailed (zoomed) information about the area covered by the tablet. The system allows multi-user interaction, although there are personalized views. Due to a specific interaction model, inputs from other devices can be recognized and displayed on other devices too. In addition there is also a vertical display capable of showing specific or detailed information about selected items. This supports the work of different groups and facilitates the around-the-table discussion.

Besides the different interaction models, there are also interaction modes based on collaboration within disaster situations. The work of Fitrianie & Rothkrantz (2007) describes an icon language (see Figure 19 and Figure 20) that can be used to describe situations and communicate them to other persons. This icon language offers an independency from language gaps as it removes all language context-information and therefore facilitates the communication within the collaboration process of team members. The icons can also be used to enrich the user interface as they describe common concepts within the disaster, for example an icon indicating fire. The different icons can be composed with a simple grammar. This grammar can be translated into natural language, so that each person can validate the correctness. Besides the support of collaboration, the icon can also be used to enrich the user interface, like e.g. maps indicating that a fire is breaking out. They can also be included in UI of different devices to give fast feedback or to annotate information (pictures, videos etc.). Further information about the ongoing work with icons as reporting mechanism can be found in (Fitrianie et. al. 2008) and (Fitrianie et. al. 2009).

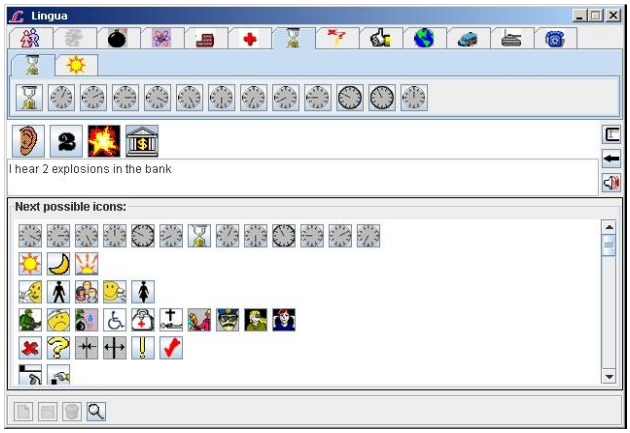


Figure 19 Icon String to communicate some observations based on (Fitrianie & Rothkrantz, 2007). Picture provided by Siska Fitrianie.

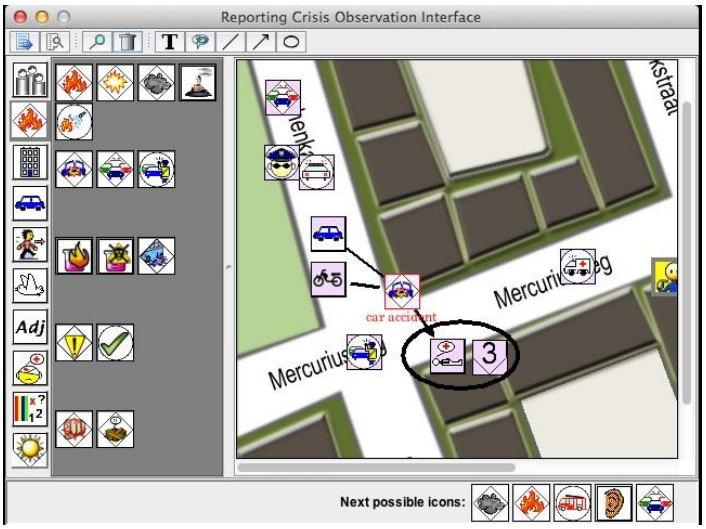


Figure 20 Message based on icon strings sent by the user on a map (Fitrianie & Rothkrantz, 2007). Picture provided by Siska Fitrianie. It shows a new version of the icon language system.

Besides the new interaction modes within HCI design, there are also simple traditional interaction patterns, relevant for traditional mouse or keyboard interaction. Below we provide a list of traditional but relevant interaction patterns (for more details about the patterns, see the Welie[15] Pattern Library):

- Action Button: Important actions that the users must be aware of, like "add to cart button".
- Guided tour: Teach the user to do something in a specific order.
- Paging: The user must navigate through a huge list of items.
- Pull-down Button: Select an action from a list of possible actions (once at a time).
- Slide show: This can also be found on mobile devices, where pictures are shown in an automatic sequence.
- Stepping: Allows the user to step through a list of items (with previous and next buttons)
- Wizard: User must make several decisions in a specific order to achieve the defined goal (like a questionnaire). For example, the questionnaire approach discussed in Section 3.2 can be seen as a wizard.

[15] http://www.welie.com/

- Enlarged Click Area: The clicking on a specific large area results in a specific view. For example clicking on a news feed shows the whole article.
- Font Enlarger: This allows the user to increase and decrease the font size as needed.

A list of relevant patterns for mobile interaction is provided in the list below. More details about these patterns can be found within the "Nokia Developer Guide" [16]:

- Indirect manipulation: Via buttons, joysticks etc. on the device
- Direct/touch interaction:
  - Long and short taps: For direct or contextual (via long taps) information.
  - Dragging: For reordering of information.
  - Sliding/Swiping: For example, the navigation on a map.
  - Spinning: A circular sliding action
  - Pinching and expanding: For example for enlarging pictures or zoom in maps.
  - Double-taps: Problematic on mobile devices (due to the waiting mode for second tap), but can be seen as double click.
- Interaction based on gestures
  - Screen orientation (portrait or landscape)
  - Device orientation (Screen up/down)
  - Change settings based on ambient light
  - Align a map based on compass orientation
  - Movements for triggering actions (like rotation)
  - Trigger an event if device is close to head, hand etc.

The work of Kervégant & Birot (2010) gives an overview of commercial products for the interaction within disaster management. Besides standard devices (smart phones, laptops, tablet PCs etc.) they also describe specialized hardware, e.g. mobile tabletops for displaying 2D and 3D maps, or large mobile projection screens. In addition, an interactive WhiteBoard beamer (eBeam) and TangiSense tabletops are introduced. For more information see the survey of (Kervégant & Birot, 2010) from the INDIGO-Project.

[16] http://www.developer.nokia.com/Community/Wiki/Mobile_Design_Patterns:_Interaction_Models

7 References

Acuna, P., Diaz, P., & Aedo, I. (2010). Development of a Design Patterns Catalog for Web-based Emergency Management Systems. *Proceedings of the 7th International ISCRAM Conference.* Seattle, USA.

Alers, H., Gunawan, L., Brinkman, W.-P., & Heynderickx, I. (2010). Effect of image quality on disaster response applications. *Second International Workshop on Quality of Multimedia Experience (QoMEX)*, (S. 41 -45).

Alexander, C. (1977). *A Pattern Language. Town, Buildings, Construction*. New York: Oxford University Press.

Alexander, C. (1999). The Origins of Pattern Theory: The Future of the Theory, and the Generation of a Living World. *IEEE Software*, 71-82.

Bader, T., Meissner, A., & Tscherney, R. (2008). Digital Map Table with Fovea-Tablett®: Smart Furniture for Emergency Operation Centers. *Proceedings of the 5th international ISCRAM Conference.* Washington, DC, USA.

Bergstrand, F., & Landgren, J. (2009). Information Sharing Using Live Video in Emergency Response Work. *Proceedings of the 6th International ISCRAM Conference.* Gothenburg, Sweden.

Carver, L., & Turoff, M. (March 2007). Human-Computer Interaction: The Human and Computer as a Team in Emergency Management Information Systems. *Commun. ACM*, 33-38.

Fitrianie, S., & Rothkrantz, L. (2009). Computed Ontology-based Situation Awareness of Multi-User Observations. *ISCRAM 2009: International Community on Information Systems for Crisis Response and Management*, (S. 13--21). Gothenburg, Sweden.

Fitrianie, S., & Rothkrantz, L. J. (Juni 2007). A Visual Communication language for Crisis Management. *The International Journal of Intelligent Control and Systems*, S. 208-216.

Fitrianie, S., Yang, Z., & Rothkrantz, L. (2008). Developing Concept-Based User Interface Using Icons for Reporting Observations. *ISCRAM 2008: International Community on Information Systems for Crisis Response and Management.* Washington DC, USA.

Flentge, F., Weber, S. G., Behring, A., & Ziegert, T. (2008). Designing Context-Aware HCI for Collaborative Emergency Management. *Workshop on HCI for Emergencies.*

Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). *Design Patterns: Elements of Reusable Object-Oriented Software.* USA: Addison-Wesley Professional.

Grill, T., & Blauhut, M. (2008). Design Patterns Applied in a User Interface Design (UID) Process for Safety Critical Environments (SCEs). *Proceedings of the 4th Symposium of the Workgroup Human-Computer Interaction and Usability Engineering of the Austrian Computer Society on HCI and Usability for Education and Work*, (S. 459--474). Graz, Austria.

Grill, T., & Tscheligi, M. (2011). Unifying Contextual Descriptions in Patterns. *5th WORKSHOP ON SOFTWARE AND USABILITY ENGINEERING CROSS-POLLINATION: Patterns, Usability and User Experience (PUX 2011)*.

Hanratty, T., Yen, J., McNeese, M., Oh, S., Kim, H.-W., Minotra, D., et al. (2009). Knowledge Visualization to Enhance Human-Agent Situation Awareness Within a Computational Recognition-Primed Decision System. *Proceedings of the 28th IEEE conference on Military communications*, (S. 2121--2127). Piscataway, NJ, USA.

Humayoun, S., Catarci, T., de Leoni, M., Marrella, A., Mecella, M., Bortenschlager, M., et al. (2009). The WORKPAD User Interface and Methodology: Developing Smart and Effective Mobile Applications for Emergency Operators. *Springer Berlin / Heidelberg*, S. 343-352.

Kervégant, C., & Birot, F. (10 2010). I4.4.2 – MOBILE DEVICES STATE-OF-THE-ART. http://indigo.diginext.fr/EN/Documents/I4.4.2-MobileDevicesSOTA.V1.3.pdf.

Krahnstoever, N., Schapira, E., Kettebekov, S., & Sharma, R. (2002). Multimodal Human-Computer Interaction for Crisis Management Systems. *Proceedings of the Sixth IEEE Workshop on Applications of Computer Vision*, (S. 203-208). Washington, DC, USA.

Micire, M., & Yanco, H. (2007). Improving Disaster Response with Multi-Touch Technologies. *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2007. IROS 2007.*, (S. 2567 -2568).

Monares, A. O., Pino, J. A., Herskovic, V., Rodriguez-Covili, J., & Neyem, A. (February 2011). Mobile Computing in Urban Emergency Situations: Improving the Support to Firefighters in the Field. *Expert Systms with Application*, S. 1255-1267.

Montells, L., Montero, S., Díaz, P., & Aedo, I. (2007). Mining Patterns for Web-based Emergency Management Systems. *Proceedings of the 7th International ISCRAM Conference*, (S. 133-138). Seatle, USA.

Nestler, S., & Klinker, G. (2007). Towards adaptive user-interfaces: Developing mobile user-interfaces for the health care domain. *In Mobiles Computing in der Medizin (MoCoMed).* Augsburg, Germany.

Newlove-Eriksson, L., & Hermansson, H. (08 2010). Symbols, Symbology and Systems: A Comprehensive Overview. http://indigo.diginext.fr/EN/Documents/I4.4.3%20Symbols,%20Symbology%20and%20Systems.pdf, EU.

Nilsson, E. G. (2009). Design Patterns for User Interface for Mobile Applications. *Advances in Engineering Software*, (S. 1318-1328). Oslo.

Norros, L., Liinasuo, M., Savioja, P., & Aaltonen, I. (30. 12 2010). *COPE - Project Publications.* Abgerufen am 15. 11 2011 von D2.3: http://cope.vtt.fi/publications.htm

Reiners, R., Astrova, I., & Zimmermann, A. (2011). Introducing new Pattern Language Concepts and an Extended Pattern Structure for Ubiquitous Computing Application Design Support. *PATTERNS 2011 : The Third International Conferences on Pervasive Patterns and Applications*, (S. 61-66). Rome, Italy.

Stiso, M. (2011). A Survey of the Users, Uses, and Perceptions of Interaction Design Patterns (manuscript in preparation)

Sutiono, A., Qiantori, A., Prasetio, S., Santoso, H., Suwa, H., Ohta, T., et al. (2010). Designing an Emergency Medical Information System for the Early Stages of Disasters in Developing Countries: The Human Interface Advantage, Simplicity and Efficiency. *Journal of Medical Systems*, S. 667-675.

Tidwell, J. (2010). *Designing Interfaces.* Canada: O'Reilly.

Turoff, M., Chumer, M., van de Walle, B., & Xiang, Y. (2004). The Design of a Dynamic Emergency Response Management Information System (DERMIS). *The Journal of Information Technology Theory and Application (JITTA)*, (S. 1-35).

Appendix 1: The BRIDGE DPL flyer

BRIDGE Design Pattern Library (D06.1)

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 patters 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 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 2: Template for collecting design patterns

Design Pattern Template

On behalf of the design pattern library team, we invite all members of the BRIDGE project to contribute with content to the library. However, as the web-based library still lacks the functionality to submit patterns online, we have put together this template for you to use in the meantime. To submit a pattern to the library you have to fill in the fields below to describe the pattern, and upload the resulting document into the folder SubmittedPatterns in the eRoom (see link in email).

Pattern name: *(Give a short and instructive name to describe the design pattern)*

Pattern type: *(Specify the type of the design pattern)*

- ○ User Interface design pattern (Choose this option if the pattern describes the graphical representation of elements in a user interface.)
- ○ Interaction design pattern (Choose this option if the pattern describes interaction with the elements of a user interface.)
- ◉ Application design pattern (Choose this option if the pattern describes the overall concept of an application type.)
- ○ Other (Choose this option if the pattern does not fall into one of the other categories above.)

Context/Usage: *(Explain in which context the design pattern is relevant)*

Problem summary: *(Please state the essence of the problem being addressed by the design pattern, using only one or two sentences.*

Solution description: *(Please give a detailed description of the solution to the problem above, including (if possible) its background, range of use and similar.)*

Illustration: *(Add an image that illustrates the solution above. This is mandatory!)*

Origin: *(Specify the origin of the pattern)*

- ○ BRIDGE: The pattern derives directly from the work within the BRIDGE project.
- ○ BRIDGE Adapted: The pattern originates from external sources, but has been adapted for the BRIDGE
- ◉ External: The pattern is derived directly from external sources.

Pattern status: *(Specify the status of the pattern)*

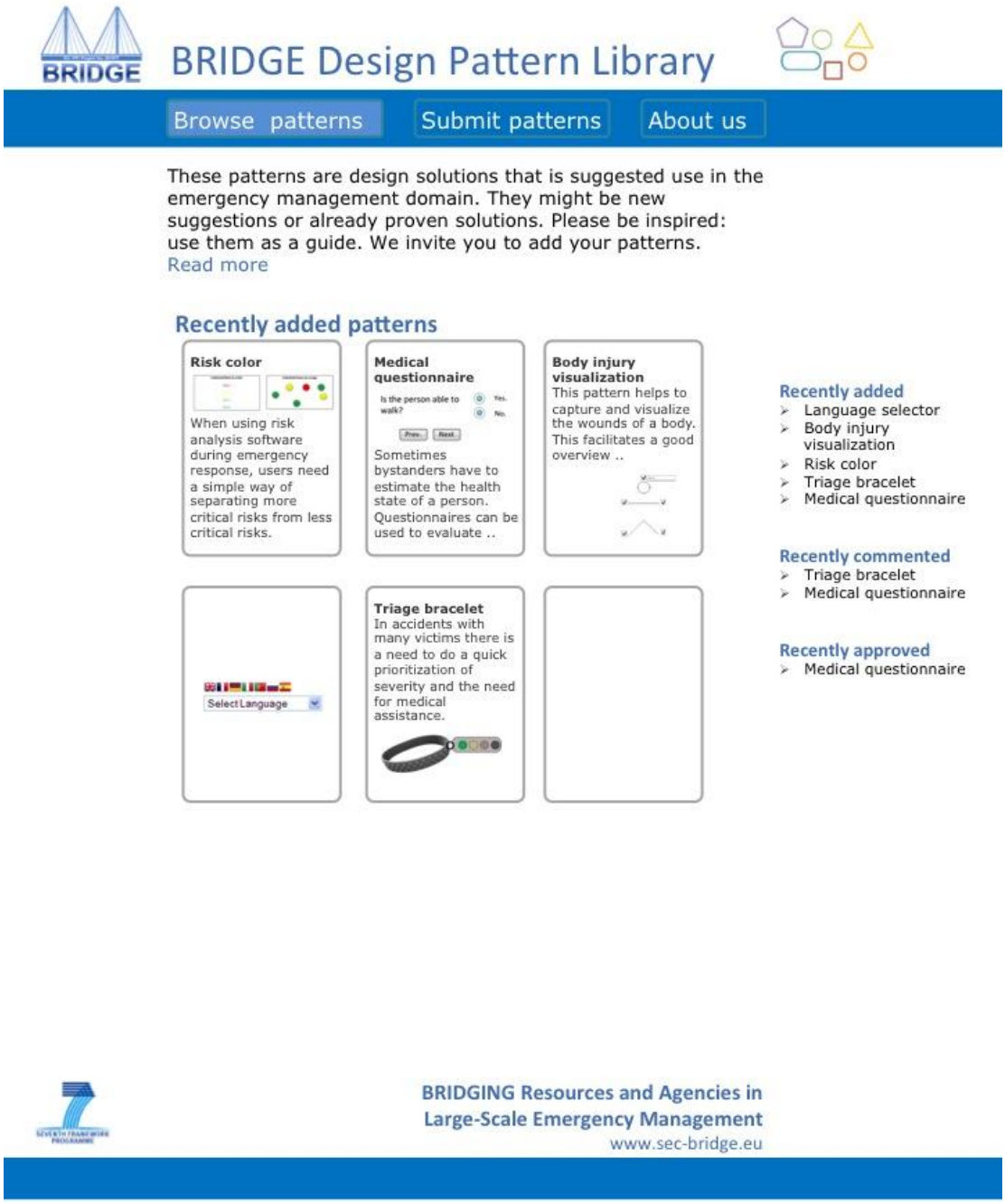
- ◉ The pattern is under consideration and has not yet been validated.
- ○ The pattern has been validated/approved.

Justification and References: *(Point to articles, reports, web pages, or similar, that confirms, mentions or describes the pattern)*

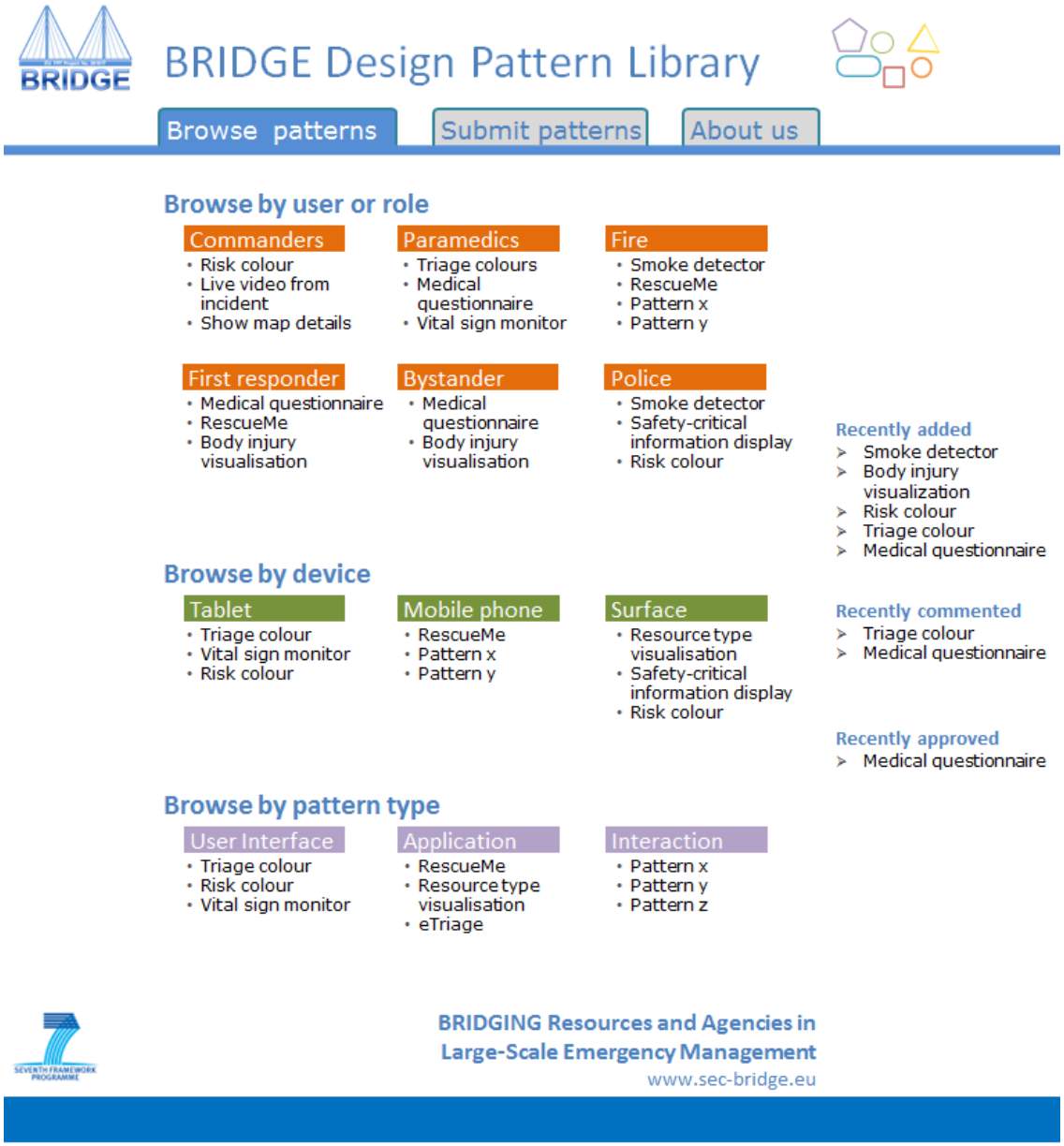
Thanks for filling in this form! If you have any questions, please send an email to WP06 *DesignPatternLibrary* from the eRoom.

Appendix 3: DPL initial design sketches

Browse patterns (all):



Browse patterns (by category):



Submit a new pattern: