Organizing User Interface Patterns for e-Government Applications

Florence Pontico[1], Marco Winckler[1] and Quentin Limbourg[2]

[1] LIIHS-IRIT, Université Paul Sabatier, 118 Route de Narbonne
31069 Toulouse, France
{Florence.Pontico, Marco.Winckler}@irit.fr

[2] SmalS-MvM, Rue du Prince Royal 102
1050 Bruxelles, Belgium
Quentin.Limbourg@smals-mvm.be

Abstract. The design of usable interactive systems is a complex task that requires knowledge and expertise on human factors and on software development. Usability guidelines and design patterns may be one way to alleviate the lack of expertise on usability of development teams by providing guidance to solve every designer's problem when designing and developing User Interface. However, the utility of guidelines and design patterns relays on two main issues: a) the quality of the advices provided, and b) the way they are organized allowing fast access to the appropriate solutions. In this paper we discuss the organization of usability guidelines and patterns at the light of an industrial project at SmalS-MvM devoted to the development of e-Government applications in a very large scale. This paper presents not only a proposal of patterns organization but also it describes a set of analysis patterns identified for e-Government applications.

Keywords: usability guidelines organization, design patterns, User Interface design process, e-Government applications.

1 Introduction

Nowadays, the problem of designing usable interactive applications has become a major concern because usability is recognized by standardization bodies like ISO [1] as a criterion of quality for software and, not less important, because poor designed application costs money to the company [2]. To study, express and ensure the usability of a User Interface, several disciplines can help every person who is responsible for developing the User Interface, notably participatory design, cognitive psychology, contextual enquiry, and software ergonomics [3]. Several methods issued from these disciplines have proven their positive impact on the usability of User Interface: usability evaluation methods with users [4], manual or automated inspection of the User Interface [5] and ergonomic approach based on guidelines [6].

In the last decades, guidelines have been used to capture and describe ergonomic knowledge. Guidelines are very versatile since they can be employed at several phases of development process. For example, they can be used to help designers to make the right design decisions and to prevent the designer from making common mistakes but also they can support the evaluation of the final product. The utility of guidelines and design patterns relays on two main issues: a) the quality of the advices provided, and b) the way they are organized allowing fast access to the appropriate solutions. In fact, many guidelines are ambiguous and can be correctly applied only by an expert on User Interface design, which creates a barrier to a wider dissemination of guidelines due to the lack of this kind of expertise in the industry [7]. It is noteworthy that even experts might experience difficulties in selecting and applying guidelines, at least in the format in which they are conflicting with one another because there is a wide gap between the recommendation (e.g., "make the web site consistent") and its applications [5].

In order to overcome this limitation of guidelines, some authors [8,9] propose to organize ergonomic knowledge under the form of design patterns. Design patterns emerged to cope with repetitive problems occurring in building architecture [10] and this concept has been extended by the Software Engineering community that created its own catalogues of proven solutions to recurrent software design problems [11]. Design patterns focus on the context of a very specific problem at a time and provide a solution that not only includes the ergonomic knowledge but also guides the designers to apply it in a practical way. Most guidelines can be extended to be expressed as patterns and the more recent research and development have preferred to present ergonomic knowledge in the form of User Interface patterns [7,12-15].

User Interface design patterns are easier to apply than guidelines but the number of patterns required to cover every usability problem increases the volume of the catalogue. This problem has already been observed when organizing guidelines [3] but it is even more dramatic in the case of patterns because patterns should be extended and reified for every application domain (e.g. web guidelines, mobile applications, etc) which increases again the volume of the information they provide.

In this paper we present a large case study conducted in the industry, at SmalS-MvM (http://www.smals-mvm.be/), where we followed the implantation of User Interface design patterns as a solution to create a usability culture in that company. SmalS-MvM is devoted to the design, deployment and handling of public e-Government applications. The discussions presented in this paper are therefore focused on e-Government domain, even though some of the lessons learned could be generalized to the organization of patterns in general. We performed an ethnographical study, which is fully described in section 2, to identify the needs in terms of access to information concerning ergonomic knowledge for the User Interface. At the light of data and evidences observed in the field, we propose, in section 3, an alternative way for organizing User Interface design patterns. During this study in the field it was possible to identify a set of patterns of User Interfaces for e-Government applications. Some of these User Interface patterns for e-Government applications are presented in section 4. In the section 5, we compare our proposal for organizing User Interface design pattern to the other organization schemas. In section 6 we discuss the lessons learned both on this state of the art of literature and on the case study we led in the field. Lastly, we present our conclusions and future work.

2 E-Government UI Analysis: a Study in the Field

SmalS-MvM is a non-profit organization devoted to the design, deployment and handling of public e-Government applications in Belgium. The current method of designing in SmalS-MvM enables the development of useful and usable e-Government applications. The design process is already user-centered, and follows many recommendations from HCI Software Engineering such as user testing and cooperative reflections led on mock-up supports onto final developed application. However, weaknesses appear about communication and reinvestment of design efforts from a project to another. That could be improved by a method of design that would fit these particular e-Government requirements.

2.1 Lots of Stakeholders, As Many Jargons and Viewpoints

One of the characteristics of e-Government is the huge number of stakeholders. This makes the design very complex because they all have to eventually cooperate in design and to be satisfied with the application while carrying different interests (interests in the application design and use, as well as political interests in general), and also different jargons, and backgrounds. Actually consider the number of persons involved in the design process:

- **Final users** can be administrative agents (social workers, office clerks, and so on) and/or citizens (individuals, representatives of an association, firm managers and firm manager secretaries). Most of the administrative procedures involving firms are actually conducted by agencies devoted to undertake procedures for the benefice of the firm. One should consider the critical aspect of the e-Government application for final users: the procedure must success because it emerges from a personal need (e.g. *I go to New-York for 2 weeks, I need a tourist Visa*) or from an administrative service need (e.g. *Visa applications have to be submitted to the embassy*), but also because personal and eventually confidential data is handled and stored during the procedure.
- **Clients** are the representatives of the institutions involved: administrative managers, commercials, domain experts and so on. The achievement of the procedure is critical for them as well, because it is intended to satisfy a need (e.g. *Management of housing benefit demands*) but also because a failure can have disastrous consequences on them in terms of corporate image. Proceedings can even be taken against the concerned institutions in some cases.
- **Design team** involves many corporate bodies for a single e-Government application: project manager, usability experts, analysts, content managers, data quality managers, graphic designers, developers, database experts, security experts and so on. They are responsible for the leading of the design process and some of them work directly with the clients (mainly the project manager, analysts and usability experts). The design firm is commercially engaged in the process which makes critical for them also that the final application permits fulfilling administrative procedures successfully and in a usable way.

2.2 Difficulties Encountered by the Design Team

SmalS-MvM employs more than 1.000 persons, mainly administrative staff, database managers, developers, architects, analysts, project managers, system and network experts. Some 25 projects are carried on, involving one or several institutions. One of the projects where many applications are developed and handled is the Social Security project. The Social Security portal (https://www.socialsecurity.be/) provides some static information and enables the fulfillment of administrative procedures in relation with the Belgian Social Security, most of them being targeted to firms. For example, the Social Risk Declaration is dematerialized on this portal: it enables an employer to declare an employee's inability to accomplish his work (e.g. in case of pregnancy, accident or disease). This way, the employee will receive allowance from the Social Security during the period he is off job. The ethnological study we led was in the context of this Social Security project.

A field observation revealed that the design process in SmalS follows many of the HCI Software Engineering recommendations. User testing is led from the very beginning of the application lifecycle, on mock-up support. User testing is done on implemented application as well. The mock-up is incrementally modified and improved until all design stakeholders agree on it. Then, the actual application (database implementation etc.) is realized and deployed. The firm is still in charge of the application after its deployment as it undertakes the call center management. Traces are kept by the call centre to allow follow-up: if a user is calling for the third time, the operator can be displayed the contents of the user's first two calls.

It appears that this iterative mock-up based process is hard to lead with so many stakeholders (see §2.1). Some weaknesses in communicating to the whole team are already noticed at the very beginning, when analysts have to transform business requirements in a first proposal to the rest of the working team. They seem to lack some expression support in order to define the application without entering into implementation details. This was quoted in a meeting, from an analyst about his work: "*I often let myself be tempted by coding some HTML pages, even if I realize that this way I already suggest design decisions that aren't yet required*". A lack of expression support is there revealed by this analyst: no tool or notation is provided, and to communicate his analysis, he uses developer's language. To cope with this lack of power of expression about recurrent topics, a UI analysis patterns catalogue is being developed towards analysts.

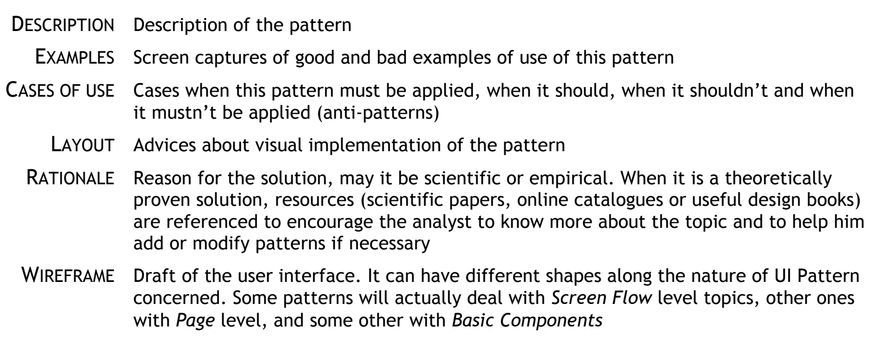
2.3 User-Centered Approach of Making Patterns

A catalogue of UI analysis patterns has to be user-centered itself, just as any application deployed that cares about being actually used. Integrating such a tool for analysts will obviously modify their way of working; however, we have to get inspired by their current design activities to make the integration as smooth and useful as possible. That is the reason why the catalogue of UI analysis patterns is made in cooperation with volunteers belonging to SmalS-MvM (mostly analysts, developers, usability experts and content managers) and who are therefore daily involved in e-Government design projects. They are not UI pattern experts, but they are interested in this initiative and, as final users of such a methodology (if not directly users of the patterns), they bring relevant comments and evaluation of the patterns in terms of their contents as well as the way to use them.

To constitute this catalogue of UI patterns, we browsed applications designed by SmalS-MvM among the ones already deployed or at advanced acceptance stages. This permitted us to pick up which UI fragments were keeping appearing in these e-Government applications. Good as well as bad examples of UI fragments were picked up in order to get as many arguments as possible for proven solutions, including by giving wrong examples (anti-patterns). Once the list constituted, those recurrent fragments of UI were integrated in UI patterns. As for the content of these patterns, we studied the design process in order to ensure a successful integration in it. Analysts are responsible for the first rough UI proposal after they have studied and treated business requirements. At CHI 2002 Workshop [16], it was suggested that wireframes could be integrated in UI patterns. This fits very well our present case: low-level fidelity UI prototypes are integrated in our UI analysis patterns, so that business requirements can be mapped to those first rough drafts.

3 Organizing UI Patterns for e-Government Applications Analysis

UI patterns can be integrated at several stages of an eGovernment design project: to support analysis and specification, to organize the information, to study graphical aspects and even to evaluate the usability of the application [14]. In this work, we focus on analysis that is the transformation of business requirements into a first specification. The specification of interaction at early stages of design is already possible thanks to several notations and formalisms, with various main intentions: supporting communication in the design team for MoLIC [17], formalizing and simulating the navigation model for StateWebCharts [18], organizing and presenting information for WebML [19]. Our own intentions are mainly the following: describe the User Interface (navigation and layout) without ambiguity though avoiding technical details, and intuitively enough so that any design stakeholder can read and at least slightly modify the description. To support these intentions, UI analysis patterns can follow the template presented in the Fig. 1.



TITLE OF THE PATTERN

DESCRIPTION	Description of the pattern
EXAMPLES	Screen captures of good and bad examples of use of this pattern
CASES OF USE	Cases when this pattern must be applied, when it should, when it shouldn't and when it mustn't be applied (anti-patterns)
LAYOUT	Advices about visual implementation of the pattern
RATIONALE	Reason for the solution, may it be scientific or empirical. When it is a theoretically proven solution, resources (scientific papers, online catalogues or useful design books) are referenced to encourage the analyst to know more about the topic and to help him add or modify patterns if necessary
WIREFRAME	Draft of the user interface. It can have different shapes along the nature of UI Pattern concerned. Some patterns will actually deal with *Screen Flow* level topics, other ones with *Page* level, and some other with *Basic Components*

Fig. 1. Template of a User Interface analysis pattern

The description of our UI patterns is rather classical (advices of implementation and rationale around a given UI design problem) until we reach the WIREFRAME attribute. Patterns have to provide solutions to recurrent problems, but it is not enough in this context: the considered solution has to be readable and understandable by every stakeholder. It even has to be a first step, an input from the analysts in the mock-up based iterative process. We therefore integrate a rough draft of the UI in our UI patterns. There are eventually different drafts illustrating several solutions to a same problem, if concrete parameters influence the application of the UI pattern. Different alternatives can be indexed in the pattern, referring to sub-patterns describing with precision each situation in which the considered sub-pattern is to be used (see MULTI-STEP WIZARD pattern Fig. 3). This way, the analyst is able to compare different propositions to choose which one better fits his proper situation. According to the level of granularity of the UI patterns, the WIREFRAME consists in a schematic representation of the layout and disposition of an UI element (ex. a page or a form), or in a rough schema of the navigation. For this former case, we chose StateWebCharts (SWC) [18] navigation modelling formalism which is an extension of StateCharts [20] devoted to navigation modeling on web applications. SWC presents the advantage of being both not ambiguous and easy to read and modify.

UI analysis e-Government patterns can be naturally classified along a hierarchical structure, following a quite traditional way of designing applications: from the general to the details. This structure implies a kind of "progressive disclosure of information" for analysts. However, when presenting them such a structure, some analysts told us about their will to have some other access to UI analysis patterns information: "*On top of that guiding procedure of browsing the UI patterns* [from top to bottom]*, I would like to be able to directly find recommendations on list boxes for example. Couldn't we have some search engine inside the catalogue?*" This request is of great interest because it outlines the need to provide direct access to patterns, in addition to top-bottom or bottom-top paths. Moreover, patterns should refer to other ones, in order to allow transversal navigation in the pyramid. Works on this topic can be found in the literature, from just considering that some patterns can 'refer to' other ones, to more complex networks using Semantic Web concepts for linking them. We will investigate afterwards in this paper (§5) existing methods to organize UI patterns to give directions on how our pyramidal organization can be completed with some relevant direct and transversal accesses.

4 Identified UI Analysis Patterns

E-Government is a highly repetitive domain, which makes design patterns relevant to reinvest design knowledge from a project to another. In particular, UI analysis patterns should be associated to the UI analysis recurrent problems listed while browsing existing applications. A support would then be provided to analysts when transforming business requirements (coming from the client) into a first draft of User Interface that will be discussable with the rest of the design team (including the client himself).

4.1 Listing of Recurrent Fragments of e-Government User Interface

The best way to list relevant UI patterns is to browse existing applications, listing manually what keeps occurring. For this activity, we browsed and studied a set of some 25 applications designed by SmalS, already deployed or in final phases of testing. Some recurrent pieces of interface stood out at three different levels of UI granularity. As a first proposal (see §3), a pyramidal structure is taken to organize patterns (see Fig. 2 below.) At the top of the pyramid, patterns stand that help structuring the application in terms of *Screen Flow*, giving directions on how to structure the procedure achievement. Underneath, interface patterns directions are given at the *Page Level*: layout of a wizard step, form fields grouping and displaying, position of the state of advancement of the procedure and so on. Lower again are *Basic Components* recommendations such as advices on how to signalize that a form field is mandatory. The basis of this pyramid is actually a set of ergonomic recommendations and even more: the "Golden Rules" recommended in HCI design whatever the application support may be [21,22].

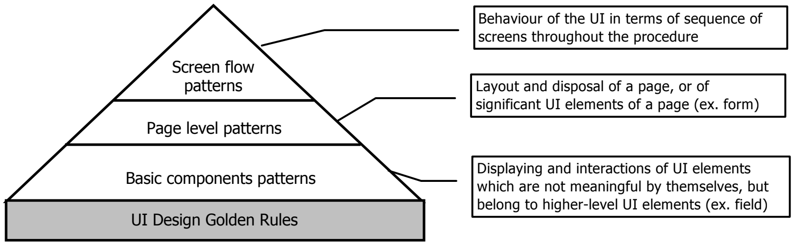

Screen flow patterns
Page level patterns
Basic components patterns
UI Design Golden Rules
Behaviour of the UI in terms of sequence of screens throughout the procedure
Layout and disposal of a page, or of significant UI elements of a page (ex. form)
Displaying and interactions of UI elements which are not meaningful by themselves, but belong to higher-level UI elements (ex. field)

Fig. 2. Pyramidal structure of e-Government UI analysis patterns

Screen Flow Level. Few sequences of pages actually occur in e-Government applications. Several of them can appear and be combined in a single application.

- *Consult and Modify Data* screen flow consists in consulting and modifying one or more items from a displayed list (e.g. management of employees' information for an employer);
- *File Management* support several activities in parallel (e.g. application for social workers to report endangered people and follow their ongoing files);
- *Hub and Spoke* flow, from a dashboard page, allows the access to a procedure just as if the user entered a funnel. At the end of the funnel, the user is led back to the first page (e.g. application for a firm employer to declare information on each employee towards the Social Security);
- *Integration in a Portal* flow is about referring and allowing access to an application from a portal (e.g. any application related to Social Security portal);
- *Multi-Step Wizard* flow consists in a strongly guided sequence of pages to achieve a single procedure (e.g. individual citizen's declaration of incomes);
- *Role Management* flow occurs as soon as the application interface depends on the role of the user (e.g. website providing offers and demands of jobs provide different functionalities to bidders and to demanders).

Page Level. Several fragments keep occurring as well at the page level. Here are some of them, possibly combined just as Screen Flow level UI fragments.

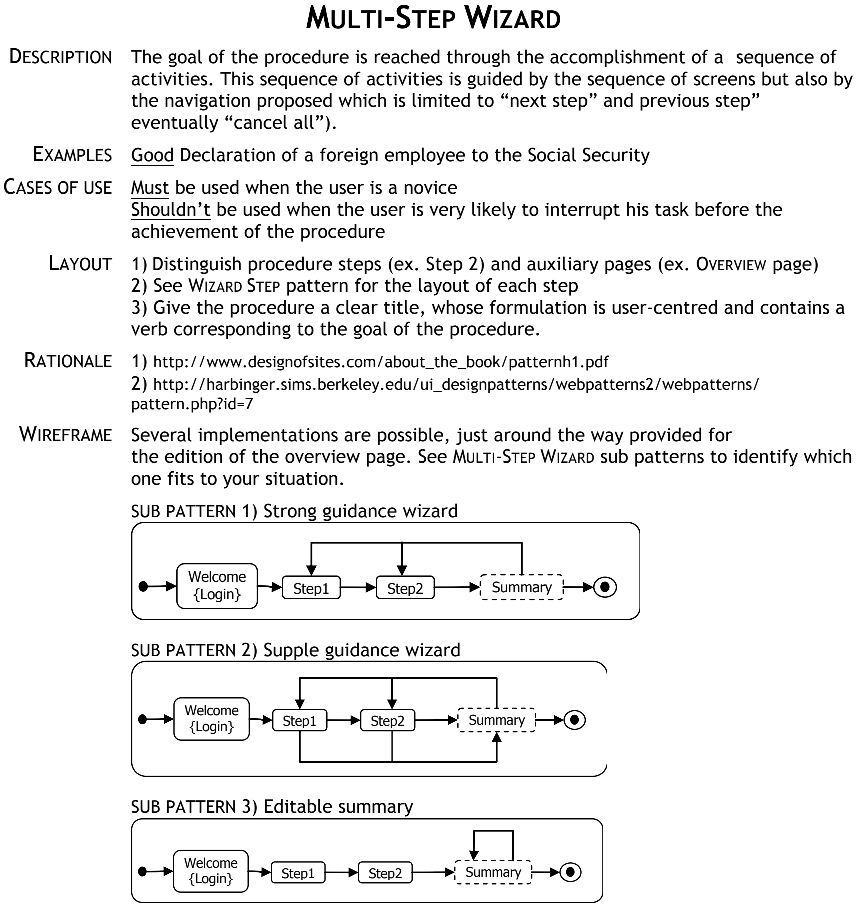
- *Acknowledgement of Receipt* page is to be displayed and proposed for printing each time a procedure has been successfully accomplished (e.g. after a Social Risk Declaration, the employees appear with the associated declarations, as well as an identifier of the web session, so that if there are some modifications to do, the declaration is easy to find);
- *Advancement Box* appears on each page of a multi-step wizard procedure to show the user his current position, what steps have been done, and which ones are to be done (e.g. during the declaration of incomes, such a box will display the sub categories of incomes to declare, and where the user is currently arrived);
- *Clear Entry Points* page supports the displaying of a few choices, each of them leading to a different part of the application, or to make the user fill in the first step of a procedure (e.g. “I want to: declare my incomes / modify my declaration / follow-up the treatment of my declaration”);
- *Filter a List* page shows how the filtering can be done and other eventual functionalities directly available on the items (e.g. for a social worker, filter should be provided on the list of cases, according to the name of the person concerned, the name of the agent who initiated the case, the date of creation, or the state of advancement of treatment of the case);
- *Overview* page is displayed at the end of the procedure and, if validated both by the user and the system, it leads to the Acknowledgement of Receipt (e.g. a summary of the Social Risks declared during the web session is displayed to the employer, so that he can check the information filled in before validating the procedure);
- *Wizard Step* has to provide the form corresponding to the current step, and some information on the state of advancement of the procedure (e.g. inheritance incomes declaration is one of the wizard steps of the incomes declaration).

Basic Components. Many fragments of the interface in terms of basic components of a page can be found in existing e-Government applications. At this level, the fragments could be applied to some close domains, such as e-Commerce for example.

- *Conditional Activation of Fields* is appreciated to deactivate the filling of a non-relevant field (e.g. “Name of the spouse?” should be deactivated in the case of a single person);
- *Download Link* have to provide information about the type of file to be downloaded, its weight and so on (e.g. proxy form, PDF format, 37 ko);
- *Mandatory Fields* have to be signalized by an asterisk just after the label (e.g. last name or social security number);
- *Non Textual Objects* such as images or video have to provide alternative text for those who can’t display them, for example blind people (e.g. “logo Social Security” as an alternative text for the picture);
- *Pre Formatted Form Fields* occur when the user has to fill a formatted field, above all when the data is intended to be automatically treated afterwards (e.g. date of birth or bank account identifier);
- *Typography* has to be taken care of, and standardized among the applications of a same portal (e.g. font size must be 11pt).

4.2 Examples of User Interface Analysis Patterns

Here are three of the UI analysis patterns that are to appear in SmalS-MvM catalogue, each one belonging to the different levels of granularity listed in section 4.1. For lack of space and for the sake of readability, bad and good examples screen captures are not displayed here. For the same reason, UI patterns are flattened: they are usually displayed as a set of tabs, with a tab for each attribute (DESCRIPTION, EXAMPLES, etc.). The first UI analysis pattern extracted from our catalogue is named "MULTI-STEP WIZARD". This is a Screen Flow level UI pattern as it describes the way a multi-step procedure should be structured among several screens when some guidance is required. This UI pattern corresponds to a very recurrent UI topic as it appears in 80% of the applications we reviewed. Three alternatives of screen flows are proposed in the WIREFRAME attribute, corresponding to different ways to let the user correct the data he filled in when he reaches the overview page. Each one of these three alternatives corresponds to a sub pattern of the MULTI-STEP WIZARD pattern (Fig. 3), as each one has to be applied in different contexts and situations.



MULTI-STEP WIZARD

DESCRIPTION The goal of the procedure is reached through the accomplishment of a sequence of activities. This sequence of activities is guided by the sequence of screens but also by the navigation proposed which is limited to "next step" and previous step" eventually "cancel all").

EXAMPLES Good Declaration of a foreign employee to the Social Security

CASES OF USE Must be used when the user is a novice
Shouldn't be used when the user is very likely to interrupt his task before the achievement of the procedure

LAYOUT 1) Distinguish procedure steps (ex. Step 2) and auxiliary pages (ex. OVERVIEW page)
2) See WIZARD STEP pattern for the layout of each step
3) Give the procedure a clear title, whose formulation is user-centred and contains a verb corresponding to the goal of the procedure.

RATIONALE 1) http://www.designofsites.com/about_the_book/patternh1.pdf
2) http://harbinger.sims.berkeley.edu/ui_designpatterns/webpatterns2/webpatterns/pattern.php?id=7

WIREFRAME Several implementations are possible, just around the way provided for the edition of the overview page. See MULTI-STEP WIZARD sub patterns to identify which one fits to your situation.

SUB PATTERN 1) Strong guidance wizard

SUB PATTERN 2) Supple guidance wizard

SUB PATTERN 3) Editable summary

Fig. 3. Example of UI analysis pattern at the Screen Flow level: "MULTI-STEP WIZARD".

Hereafter, Fig. 4 presents the ADVANCEMENT BOX UI pattern. It belongs to the Page Level as it concerns the layout, disposal and behavior of a UI fragment which has a sense by itself in the application. This pattern appears (or should appear) in as many applications as the MULTI-STEP WIZARD pattern we saw above, which means very often in e-Government applications.

ADVANCEMENT BOX

DESCRIPTION Display the user its current position in the procedure: where he is, what he has done successfully or not), what is left to be done.

EXAMPLES Good Declaration of a foreign employee (box on the right of the screen)
Bad Declaration of socially endangered persons (no advancement box)

CASES OF USE Must be used in multi-step wizard procedures holding three or more steps

LAYOUT 1) Use 2 shades of 1 colour for the background of the box places, the deeper one signalizing the current step, the lighter for the steps done or to be done.
2) Use three icons to show the state of a step (e.g. ✓×~)
3) Don't use checkboxes to indicate (completed) steps as this can give a false impression users can click on them.
4) Give each step a number
5) Put the box in the right-hand part of the screen, just as any non critical information that can be missed by people holding a low screen resolution.

RATIONALE Van Welie "Purchase Process" pattern is close to this one, with a line instead of a box
http://www.welie.com/patterns/showPattern.php?patternID=purchase-process

WIREFRAME The advancement box appears on the right-hand side

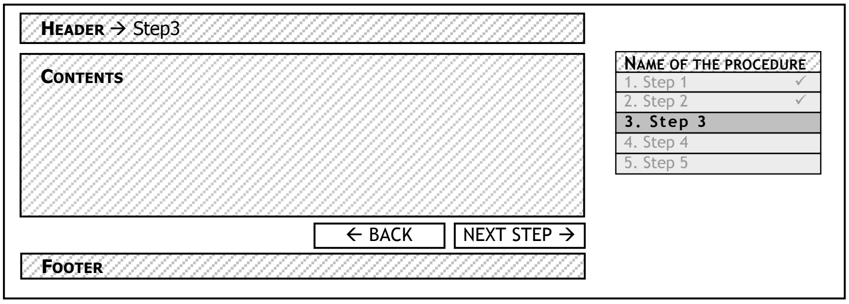


Fig. 4. Example of UI analysis pattern at the Page level: "ADVANCEMENT BOX".

MANDATORY FIELD

DESCRIPTION Warn the user about the fields required to pursue the procedure

EXAMPLES Good Forms of the social workers' application supporting cases management
Bad Forms of the incomes declaration (bad disposition of the asterisk)

CASES OF USE Must be used as soon as there are mandatory AND optional fields in a form.

LAYOUT 1) Use an asterisk, just after the label of the concerned field
2) Write an obvious legend
3) Insert an asterisk (character) or an image

RATIONALE http://www.welie.com/patterns/showPattern.php?patternID=forms.

WIREFRAME The third field is mandatory in this example

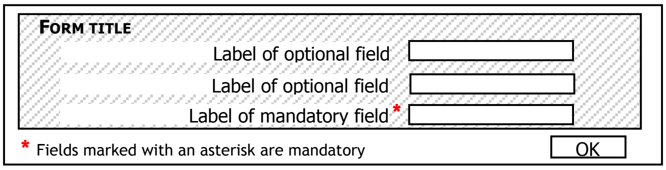

FORM TITLE
Label of optional field
Label of optional field
Label of mandatory field *
* Fields marked with an asterisk are mandatory
OK

Fig. 5. Example of UI analysis pattern at the Basic Component level: "MANDATORY FIELD".

Our last example is presented above in Fig. 5 and is called "MANDATORY FIELD". This Basic Component UI pattern could appear in any web application holding forms and caring for usability. This pattern is useful because, if most of the applications investigated do signalize the mandatory fields, many of them don't place correctly the asterisk just after the label which is yet better for the readability of the form. In other terms, this is the kind of UI patterns that carries usability principles which are basics ones but often missing. It outlines as well that our UI analysis patterns catalogue strongly suggests a uniform solution to analysts. Other ways to distinguish mandatory fields could actually have been suggested (ex: *use red to label mandatory fields* or just some *distinguish mandatory fields from optional ones* advice) but our purpose here is to provide directly applicable and unified solutions to analysts, towards uniformed e-Government applications, at least for the applications belonging to the same portal, such as in our case with the Social Security portal.

5 Related Work

One of the major issues for the use of User Interface patterns in the practice is the proper organization of patterns in accessible catalogues providing fast access to the appropriate solutions. Fincher [8] claims that patterns must be organized in such a way that they are easy to locate, they are grouped when appearing in common cases, they provide different viewpoints, and they permit to generate new solutions from the ones proposed. The most famous collections of UI patterns provide some intrinsic classification that is a proposal of some categories supposed to be useful for an efficient browsing of the collection. These catalogues might concern User Interfaces in general [12,24] or be focused on a particular application domain such as web applications [25], e-Commerce [26,27], and mobile applications [28]. Specialized catalogues are created by selecting already known patterns and, based on experience on considered field, adapting known patterns and identifying new ones. As far as we know, there is not yet a catalogue for e-Government applications. This might be explained by the emergence of e-Government and as such, some time is needed for the community to identify successful solutions that could be clearly stated as *patterns*.

5.1 Currently Available UI Patterns Catalogues and Inner Organization

Hereafter we present a short summary of most representative UI patterns catalogues found in the literature. We focus in particular on the way the patterns are organized in the catalogue rather than their content.

The **Van Welie's catalogue** [29] is a large catalogue which is organized in subsets according to the application domain: ex. Web-based applications, mobile applications and GUI design in general (which is at a higher level of implementation detail for the design phase we consider here that is early UI specification). In this catalogue, patterns are basically centred on the user's intentions. Examples of categories and patterns in categories: *SITE TYPES (ex. artist site, portal, etc.), USER EXPERIENCES (ex. fun, shopping, etc.), E-COMMERCE (ex. shopping cart, store locator, etc.)*, etc.

The **Yahoo! Design Patterns Library** [13] follows a goal-oriented approach. Reflections on how authors came to this classification are available online [30]. The outlined goals actually include user's goals and designer's goals, considering that the User Interface has to satisfy both of them. This way, user's intentions and needs can be satisfied – for example: ***USER NEEDS TO:*** *NAVIGATE (ex. of patterns: breadcrumbs, tabs, etc.), EXPLORE DATA (ex. calendar picker, pagination, etc.)…* – as well as designer's technical constraints – for example: ***APPLICATION NEEDS TO:*** *CALL ATTENTION (ex. help by dynamic tool tip, transition with an animation, etc.), GROUP RELATED ITEMS (ex. scrolling list, tree, etc.), etc.*

The **Coram's catalogue** introduced Experiences [31] as a new UI pattern language in order to cope with high-level UI design problems. These patterns are grouped by focus and belong to a network which is presented in Fig. 6. From "Interaction style" meta-pattern, patterns are grouped and linked in four categories, corresponding to how the user is intended to interact with the application.

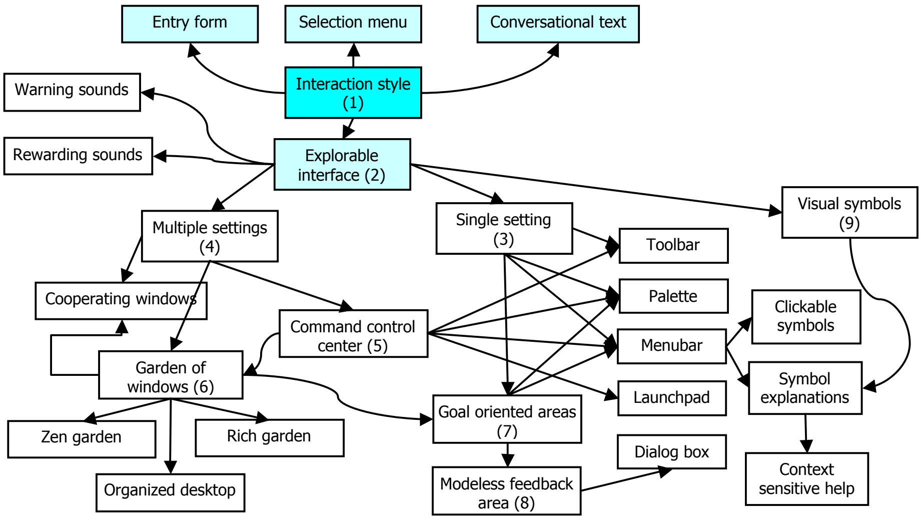


Fig. 6. Partial view of the map of the Experiences UI patterns (from [31])

The **Laakso's catalogue** [24] covers several kinds of applications, including web but not only. Most is done about visualisation that is about how information and/or data are organized (*ex. DATA VIEWS category contains these patterns: overview beside detail, fisheye, etc.*) even though some categories are devoted to displaying of information (*ex. TIME: calendar strip, schedule*); command interactions are included as well (*ex. SAVE AND UNDO: auto-save, object-specific undo, etc.*).

The **Tidwell's catalogue** [12] is a collection of generic UI design patterns that can be used to deal with web applications, mobile applications or any other kind of interfaces. The patterns are very generic and cover multiple levels of the User Interface design. Some of the categories are entirely devoted the description of interactions with users (*ex. category GETTING INPUT FROM USERS, contains the following patterns: forgiving format, dropdown chooser, etc.*). Other examples of patterns categories: *ORGANIZING THE CONTENT (ex. of patterns: two-panel selector, wizard, etc.), SHOWING COMPLEX DATA (ex. overview plus detail, cascading lists, etc.)*, etc.

The **Van Duyne's catalogue** [23] is designer-oriented (e.g. "*helping customers complete tasks*") but the catalogue aims to follow a "customer-oriented approach". This calling emphasizes the help that is given about functional and procedural aspects of the web application, such as "buying products" or "search for a similar product". At the beginning, there is some progressive in-depth display of the patterns (site genre, then navigation framework, then homepage), but it is lost afterwards, in favour of more general advices. Example of categories in this catalogue: *SITE GENRES (ex. of patterns: personal e-Commerce, self-service government, etc.), CREATING A NAVIGATION FRAMEWORK (ex. alphabetical organization, popularity-based organization, etc.), CREATING A POWERFUL HOMEPAGE (ex.: homepage portal, up-front proposition),* etc.

The **Montero's catalogue** [25] aims to guide design towards usable web applications. Its specificity is that patterns in this catalogue are grouped along three levels of abstraction: *WEB SITE*, *WEB PAGE* and *ORNAMENTATION*, based on Alexander's first works about architecture patterns [10]. Moreover, a network weaves patterns throughout categories, around common ergonomic advises for web design, as it is shown in the Fig. 7 below.

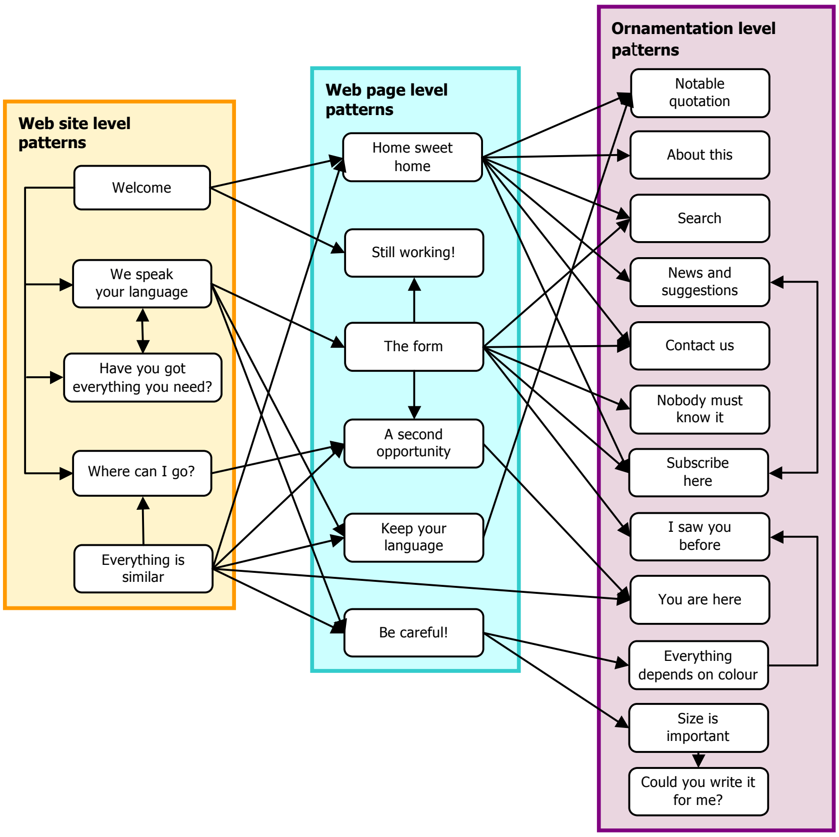


Fig. 7. Montero's proposed pattern language (from [25])

In addition to these catalogues, some authors provide alternative methods for structuring their catalogues. The rationale behind this alternative organization is that patterns could therefore be "composed of" or "derived from" another one or other ones. Two classification proposals following such an approach are noteworthy: the **Object-Oriented organization** proposed by Van Welie et al. [7], and the proposal of Henninger et al. [15] using a **Semantic Web** approach.

Van Welie et al. [7] investigated the possibility of structuring web UI patterns in a hierarchical way featuring an Object-Oriented organization. This way of structuring with a top-down approach is actually similar to what had been proposed at the very beginning of the design patterns history [10]. Web design patterns can therefore belong to different levels that are:

- POSTURE: reason for existence of the application (ex. *e-Commerce*, *Personal site*), coming from the business goals (ex. *Customer satisfaction*, *Selling products*);
- EXPERIENCE: high level goal for which the user comes to the website, beyond functional tasks and goals (ex. *Playing*, *Shopping*, *Browsing*, *Sharing thoughts*);
- TASK: solutions to small user problems which are part of a higher level "Experience"; the solution is given in terms of a set of interactions (ex. *Product comparison*, *Identify*);
- ACTION: pieces of interaction that are at the lowest level of interest for UI patterns; they are meaningful only if they are related to a task pattern (ex. *Pushbutton*).

Moreover, some precisions are given that imitate Object Oriented modeling, in order to distinguish different relationships of connecting patterns: aggregation, specialization and association.

The approach suggested by Henninger et al. [15] aims to make more pro-active the representation of sets of design patterns in general (i.e. not especially UI patterns, but we only observe the structure principle here, which could be applied to UI patterns). The authors presuppose that weaving design patterns thanks to Semantic Web methods would provide a more usable and navigable set of patterns for designers. As Semantic Web is developed to cope with a more efficient and supple access to information whose volume is increasing, this method may effectively be of interest for design patterns information. A tool is associated to this framework, supporting the edition of ontology to weave the design patterns: BORE (Building an Organizational Repository of Experiences) [32]. The main goal of this section is rather discuss strategies for guidelines rather than the content of patterns themselves.

5.2 Classifying UI Catalogues Organizations

Based on our experience in the field at SmalS-MvM, we have identified some suitable requirements for organizing UI patterns. Hereafter we present a list of these requirements, which were inspired from Fincher's criteria, to evaluate the organization of currently available UI patterns catalogues:

- **Hierarchical/Pyramidal** access has to be provided as a "progressive disclosure of information" which is a natural way of thinking design.

- **Cross References on UI Elements** appearing in different patterns. For example, if a pattern contains a list box, references should be available to other patterns in which list boxes appear as well.
- **Siblings grouping.** Patterns which often are of interest in common cases should be put together and therefore create similar families' patterns that may be applied to similar applications.
- **Viewpoints comparison.** In some cases, several patterns can be applied. This criterion is about the way the designer is supported in this choice.
- **Evolution and scaling**. Can the list of existing patterns be augmented? Is scaling possible? In other terms, this criterion tells if the investigated organization of patterns would bear an important volume of data.

The Table 1 provides a comparison of patterns catalogues found in the literature according to the criteria aforementioned.

Table 1. Evaluation of reviewed organizing UI patterns principles

	Pyramidal structure	Cross-ref on UI elements	Siblings grouping	Viewpoints comparison	Evolution and scaling
Coram	✓	×	✓	✓	×
Henninger	×	✓	✓	✓	✓
Laakso	×	×	×	×	×
Montero	✓	×	×	×	×
Van Duyne	✓	×	✓	✓	~
Van Welie catalogue	✓	×	✓	✓	×
Van Welie oo organization	✓	×	✓	✓	~
Yahoo!	~	×	✓	✓	×

Legend: ✓ supported, × not supported, ~ cumbersome

6 Lessons Learned

Integrating some new artifact as a support to an existing activity is a sensitive process. The way of leading the activity will anyway have to be adapted to this new artifact, whatever its quality will be. For a supple adaptation, the authors of the artifact have to consider how users currently carry activities, and, as much as possible, to confront the project of artifact to their opinion. If not, the artifact is very likely to be rejected (in the case of a commercial product for example) or diverted by its users towards a way that better fits their habits and needs (in the case of a support to work for example). Observation and user testing are therefore wise ways to design useful and usable products. To follow this HCI basic principle, we had to learn more about analysts' activities both from current web design methodologies [33-35] and from analysts' observations and reactions to the UI patterns proposed. In parallel, to feed our reflection on UI patterns, we reviewed the literature and studied other works' experiences and conclusions. This section is a summary of the lessons learned both from the ethnological study and the UI patterns literature browsing.

Need for e-Government patterns. The browsing of existing e-Government applications revealed that patterns are relevant for e-Government which is a highly repetitive context, with many recurrent fragments appearing (see §4.1 for the particular case of UI patterns). Moreover, the strong rationale included (by definition) in UI patterns would help coping with some decisions that may be hard to take when stakeholders hold divergent interests. Patterns help bringing people's opinions back together for the benefice of the application, which is very useful in e-Government where so many stakeholders are involved (see §2.1).

Need for e-Government specific UI patterns. UI patterns have to be close to their application domain. In particular at the highest level, specific UI fragments occur as we consider a defined domain. E-Commerce UI patterns are proposed in several studies such as Van Welie's catalogue [29], which can somehow but not entirely help building e-Government applications, in spite of their common points [27]. For example, an e-Commerce Page Level UI pattern includes incentives to buy additional products whereas in e-Government, the purpose is to provide clear and formal information to fulfill the goal, not to give rise to new wishes.

User-centred UI patterns and catalogue. Integrating a design support must be done with respect to current activity. The UI patterns catalogue must therefore be user-centred. Observations and meetings with design team members, as well as investigations on theoretical design practices are done all along the making of this catalogue (see §2.3). However, some rigorous user testing has to be carried out as soon as the catalogue is complete enough to be operational.

E-Government UI patterns content. The usability of the UI patterns proposed first depends on their content (see §3). Bad and good examples have to appear to support and illustrate the rationale included in the patterns. Both static and dynamic aspects of the application have to be described in a non ambiguous though "easy to read and modify" way. The static behaviour mainly refers to the layout of the pages and UI elements (¡such as forms). UI patterns on static topics are accompanied by wireframes to be an efficient support for communication among stakeholders. For the same reason of readability and non ambiguousness, StateWebCharts formalism ensures the representation of the dynamic aspects (navigation among the application).

E-Government UI patterns organization. UI patterns have to be displayed in a way that suggests their actual use. The investigations we made in the field revealed that analysts not only need a progressive disclosure of information, but also some transversal access to the UI patterns information (§3). UI patterns organization has therefore to be efficient concerning easy location of patterns, cross references on UI elements or on context of application, comparison and grouping of patterns applicable in close situations, and finally a possible increasing of the number of UI patterns while keeping the benefits of the organization. Existing methods of UI patterns organization don't fit these requirements, globally failing in providing relevant cross references among patterns, and in supporting an evolution that would lead to a huge number of patterns (§5.2). However, Semantic Web principles appear to be the more relevant among the organization principles investigated.

7 Conclusion and Future Work

As e-Government influence keeps increasing, more and more IT firms are eager to invest their efforts into this complex domain. The important number of stakeholders involved in such projects makes e-Government design a hard activity to lead. They are critical systems for the institutions involved as well as for final users. To ensure that the goals of these final users will be satisfied thanks to a usable application, UI patterns are a solution. We studied contents for e-Government UI patterns as well as an organization for a user-centered displaying of UI patterns to analysts. This study was based on an ethnological study as well as on literature. These investigations prompted us to find a relevant organizing UI patterns as a critical topic for UI patterns usability and acceptance in the design team. The UI analysis patterns catalogue contents and organization are strongly related to the activity observed in the field and also to the particular tasks fulfilled in the investigated domain. Users and their supposed tasks are well-known in this mature e-Government domain, and that was the basis of the catalogue building. This is actually a limit of our work which, to be extended to other domains, would necessitate the same kind of investigation in the field and inventory of recurrent patterns. However, this methodology employed to build UI analysis patterns could be reinvested for other domains, in particular the lessons learned about the development of a user-centered organization of patterns and their integration in a design process. Our future work envisages the building of an ontological mapping of the concepts appearing in UI patterns. Inspired by Semantic Web principles, this could support as many navigation links among UI patterns as there are links among UI patterns concepts. Moreover, by nature, this kind of structure would support the enlargement of the existing UI patterns catalogue. The necessary support to consult and edit patterns has to be considered as well. This possibility may be given as well for the user of the catalogue to make his customized organization. UI patterns are a relevant support for e-Government design because it copes with recurrent design questions with a strong rationale and first proposals towards a usable application.

Acknowledgments. This was possible thanks to SmalS-MvM and to COST294-MAUSE (http://www.cost294.org/).

References

1. ISO/WD 9241, "Ergonomic requirements for office work with visual displays units" International Standard Organization, 1992.
2. Mayhew, D. J. and Bias, R. G. *Cost-justifying usability*, Morgan Kaufmann, 1994.
3. Vanderdonckt, J., Development milestones towards a tool for working with guidelines *Interacting with Computers*, vol. 12, no. 2, pp. 81-118, 1999.
4. Hix, D. and Hartson, R. *Developing user interfaces: ensuring usability through product and process*, New York, USA: John Wiley & Sons, 1993.
5. Ivory, M. Y. *Automated web site evaluation: researchers' and practitioners' perspectives*, Dordrecht, The Netherlands: Kluwer Academic Publishers, 2003.

6. Bastien, C. and Scapin, D. L., Evaluating a user interface with ergonomic criteria *International Journal of Human-Computer Interaction*, vol. 7, pp. 105-121, 1995.
7. Van Welie, M. and Van der Veer, G. C., "Pattern languages in interaction design: structure and organization" *Interact 2003,* Zürich, Switzerland, 2003.
8. Fincher, S. and Windsor, P., "Why patterns are not enough: some suggestions concerning an organizing principle for patterns of UI design" *Workshop Pattern languages for interaction design: building momentum (at CHI2000),* The Hague, The Netherlands, 2000.
9. Javahery, H. and Seffah, A., "A model for usability pattern-oriented design" *TAMODIA,* Bucharest, Romania, pp. 104-110, 2002.
10. Alexander, C., Ishikawa, S., Silverstein, M., Jacobson, M., Fiksdahl-King, I., and Angel, S. *A pattern language*, New York, USA: Oxford University Press, 1977.
11. Gamma, E., Helm, R., Johnson, R., and Vlissides, J. *Design patterns, elements of reusable object-oriented software*, Reading, Massachussets: Addison Wesley, 1995.
12. Tidwell, J. *Designing interfaces*, Sebastopol, California, USA: O'Reilly, 2005.
13. Yahoo! Design pattern library. http://developer.yahoo.com/ypatterns/ . 2006.
14. García, F. J., Lozano, M. D., Montero, F., Gallud, J. A., González, P., and Lorenzo, C., "A Controlled Experiment for Measuring the Usability of Webapps Using Patterns" *ICEIS,* Miami, USA, 2005.
15. Henninger, S. and Ashokkumar, P., "An ontology-based infrastructure for usability patterns" *Workshop on Semantic web enabled software engineering (at ISWC),* Galway, Ireland, 2005.
16. Van Welie, M., "Patterns for designers?" *Patterns Workshop (at CHI),* Minneapolis, USA, 2002.
17. Greco de Paula, M., Santana da Silva, B., and Diniz Junqueira Barbosa, S., "Using an interaction model as a resource for communication in design" *CHI* , Portland, USA, 2005.
18. Winckler, M. and Palanque, P., "StateWebCharts: a formal description technique dedicated to navigation modelling of Web applications" *DSV-IS,* Madera Island, Portugal, 2003.
19. Ceri, S., Fratemali, P., and Bongio, A., "Web Modeling Language (WebML): a Modeling Language for Designing Web Sites" *WWW9 Conference,* 2000.
20. Harel, D., Statecharts: A visual Formalism for complex systems *Science of Computer Programming*, vol. 8, no. 3, pp. 231-274, 1987.
21. Shneiderman, B. *Designing the user interface. Strategies for effective human-computer interaction*, Addison-Wesley, 1998.
22. Constantine, L. L. and Lockwood, L. A. D. *Software for use*, New York, USA: ACM Press/Addison-Wesley Publishing Co., 1999.
23. Van Duyne, D. K., Landay, J. A., and Hong, J. I. *The design of sites*, Addison-Wesley Professional, 2002.
24. Laakso, Sari A. User interface design patterns. http://www.cs.helsinki.fi/u/salaakso/patterns/ 2003.
25. Montero, F., Lozano, M., González, P., and Ramos, I., "Designing websites by using patterns" *SugarLoafPLoP,* Itaipava, Brasil, pp. 209-224, 2002.
26. Rossi, G., Lyardet, F., and Schwabe, D., "Patterns for e-Commerce applications" *EuroPLoP,* Irsee, Germany, 2000.
27. Wimmer, M. A. Knowledge Management in e-Government – e-Commerce vs. e-Government. http://falcon.ifs.uni-linz.ac.at/research/ceepus.zip. 2001.
28. Van Welie, M. MobileUI design patterns. http://www.welie.com/patterns/mobile/ 2007.
29. Van Welie, M. Web design patterns. http://www.welie.com/patterns/ 2007.
30. Malone, E., Leacock, M., and Wheeler, C., "Implementing a Pattern Library in the Real World: A Yahoo! Case Study" *ASIS&T,* Montréal, Canada, 2005.
31. Coram, T. and Lee, J. Experiences – A pattern language for user interface design. http://www.maplefish.com/todd/papers/Experiences.html 2002.

32. Henninger, S., Ivaturi, A., Nuli, K., and Thirunavukkaras, A., "Supporting adaptable methodologies to meet evolving project needs" *Joint conference on XP universe and Agile universe,* Chicago, Illinois, USA, pp. 33-44, 2002.
33. Ivory, M. Y. and Hearst, M. A., Improving web site design *IEEE Internet Computing*, vol. 6, pp. 56-63, 2002.
34. Nogier, J.-F. *De l'ergonomie du logiciel au design des sites Web*, Paris, France: Dunod, 2002.
35. Nielsen, J. Alertbox. http://www.useit.com/alertbox/ 2006.

Questions

Phil Gray:

Question: Did you include "Forces" in your pattern description?

Answer: Yes, it is within the RATIONALE section of the pattern. Whenever possible, references to academic papers or to other pattern catalogues are given. This way, the user of our catalogue of patterns can know more, extend his study and eventually the catalogue of patterns.

Question: Did you define relationships between patterns that, for example, point to patterns that propose alternative solutions for similar problems?

Answer: Yes, links to related patterns are included in the pattern description. A global map of links between patterns is being developed, to allow navigation among the patterns which allows in particular the comparison of patterns between them. This mapping is an ontological mapping of the concepts appearing in the patterns.

Ann Blandford:

Question: What makes your patterns specific to e-government applications?

Answer: The patterns we present here have been discovered while working for an e-Government enterprise. Hence they are applicable for, but not limited to, the domain of e-Government applications. Some of the patterns are also applicable in a broader context, for example the ones describing the behaviour of UI elements (cf. the MANDATORY FIELD pattern exposed in Fig.5). By the way, a notion of standardization is included in our patterns: this is acceptable for e-Government, for the sake of UI coherence across different applications, to let the user adapt to the UI one time for all the times he will visit a governmental website. Due to marketing reasons, it would be very difficult to create uniform user interfaces in other domains such as e-Commerce.