A Design Process for Exhibiting Design Choices and Trade-Offs in (Potentially) Conflicting User Interface Guidelines

Llúcia Masip, Célia Martinie, Marco Winckler, Philippe Palanque, Toni Granollers, Marta Oliva

A Design Process for Exhibiting Design Choices and Trade-offs in (potentially) Conflicting User Interface Guidelines

Llúcia Masip[1], Célia Martinie[2], Marco Winckler[2], Philippe Palanque[2], Toni Granollers[1] and Marta Oliva[1]

[1] DIEI (Universidad de Lleida)
c/Jaume II, 69, Lleida 25001, Spain
{lluciamaar, tonig, oliva}@diei.udl.cat

[2] IRIT, Université Paul Sabatier
118 route de Narbonne, 31062 Toulouse Cedex 9, France
{winckler, martinie, palanque}@irit.fr

Abstract. In the last decades a huge amount of knowledge about user interface design has been gathered in the form of guidelines. Quite often, guidelines are compiled according to user interface properties (e.g. usability, accessibility) and/or application domains (e.g. Web, mobile). In many situations designers have to combine several guideline sets in order to address the specific application domain and the desired set of properties corresponding to the application under consideration. Despite the fact that the problems related to the selection of guidelines from different sources are not new, the occurrence and management of conflicting guidelines are poorly documented leaving designers with little help in order to handle conflicts in a rationale and consistent way. In this paper we revise the questions related to selection and management of conflicting guidelines and we propose a systematic approach based on design rationale tools and techniques for exhibiting choices and trade-offs when combining different guidelines sets. This paper illustrates how such as an approach can also be used to deepen the knowledge on the use of user interface guidelines recording decisions across projects in an iterative way.

Keywords: user interface guidelines, guidelines management, design rationale.

1 Introduction

The design of usable interactive systems is a complex task which requires knowledge and expertise both on human factors and software development. User interface guidelines represent a technical solution for organizing recommendations and best practices which are aimed at providing guidance throughout the development process of interactive systems [1]. User interface guidelines are widely available but, quite often they are in many different formats with contents varying both in terms of quality and level of details. Recommendation for User Interface development appear in the literature under various names including principles for user interface design, heuristics, guidelines, user interface design patterns and standards. Whilst principle

and heuristics (such as "*give appropriate feedback*") tend to be the least prescriptive and general, design standards (such as ISO [16]) are very specific, though also very restrictive and prescriptive. User interface guidelines and design patterns lie in between these extremes. Design patterns, in particular, offer invariant solutions to a recurrent problem within a given context [2]. In general, such recommendations (either standards, guidelines…) appear together in the literature as a set of interlinked recommendations for a specific domain [3][4]. For the sake of simplicity, this paper exploits the term 'user interface guidelines' with its most generic meaning but our approach can also be used so resolve conflicts with the other kinds of recommendations.

The Human-Computer Interaction (HCI) community has been prolific in the development of guidelines for interactive systems [5]. These recommendations are usually gathered in compilations which are often organized by user interface properties (e.g. usability, accessibility, user experience) and/or by application domains (e.g. Web, mobile applications, tabletops). Currently there is a wide range of sources of guidelines available including many HCI areas such as web-based systems [6], safety critical systems [7], cooperative interaction [8], ubiquitous computing [3], interactive TV [8], web accessibility [9], UX patterns [3]…

In many situations designers have to combine different guidelines sources in order to address the specific application domain and the desired set of properties corresponding to their project [5]. For instance in the Ubiloop project we make use of mobile and Web technology for allowing citizens to report urban incidents. As there is no user interface guideline compilation covering Web and mobile technology, incident reporting systems, usability, security, and user experience, we had to combine different guideline sources. The combination of different guidelines ended-up with a huge list of entries containing duplicated entries, similar statements using different terms, guidelines that refer to elements that are not relevant to the project, and potentially conflicting guidelines (for example, security guidelines recommending validation steps that contradicts with usability guidelines that recommend minimal actions). In order to design a user interface meeting both usability and security in such context, a cleaning-up selection process was required to provide reliable, consistent and usable set of guidelines.

These problems related to the selection of guidelines from different sources have been previously reported in the literature (such as in [10] and [11]) and motivated the development of tools for working with guidelines [12] and [5]. Nonetheless, the inner problems related to the occurrence of potentially conflicting guidelines have been poorly documented so far. The resolution of conflicts is a daunting and demanding task that often requires taking into account the trade-offs associated with alternative design choices. Therefore, whenever a good solution for solving conflicts between guidelines is found, it is worth the effort recording and documenting it for further reuse. This is the contribution of this paper which proposes a systematic approach for selecting guidelines and documenting the conflicts managements. Hereafter, design rationale is proposed as a complimentary technique that can be ultimately included as an integral part of the tools for working with guidelines. Our ultimate goal is to integrate the design process presented in the current paper into a tool (in line with the Open-HEREDEUX project [13]) for supporting the detection of conflicts between guidelines, the resolution of conflicts and the reuse of previously defined solutions.

The paper is organized as follows: section 2 provides an overview of existing approaches and tools for working with guidelines; section 3 describes the how design rationale can be used to solve conflicting guidelines; section 4 provides an overview of our design process including the integration of support for design rationale; section 5 illustrates the approach through a case study involving conflicting guidelines. Lastly, section 6 concludes the paper and identifies directions for future work.

2 Overview of User Interface Guidelines Management

This section points out the main questions related to user interface guidelines management. For a full survey please refer to [5] and [37].

2.1 Organization of guidelines sources

One of the major issues for the effective use of user interface guidelines is the fast access to the appropriate design solutions [4]. Guidelines must be organized in such a way that they are easy to locate, that they are grouped when appearing in common cases, that they provide different viewpoints, and that they permit to generate new solutions from the ones proposed. Several works focus on the organization of guidelines for improving search of guidelines in large datasets [30] and many others try to organize guidelines in a way they can become easier to understand and apply along the different phases of the development process [14]. Some works [3][15] propose XML-based languages for structuring the description of guidelines.

Currently, there are many sources of user interface guidelines which are organized by the intended use of guidelines (e.g. support design, development and/or testing phases in the development process), the level of formalization of guidelines sets (user interface design guidelines/design patterns, standards…) and/or the scope (ex. generic guides, platform-specific, corporation-specific style guides, etc.) [5].

Participants on the development process (i.e. task analyst, the project leader, the human factors expert, graphical designers, the designers, the programmers and the user interface evaluators) might be interested in particular sub-sets of guidelines. Furthermore, some years ago, the only factor which was considered when a designer or evaluator wanted to design or evaluate an interactive system was the usability [16]. But to now factors such as accessibility [9], communicability [17], cross-cultural [18], plasticity [19], playability [20], security [21] and user experience [22][23], become important elements for deciding the quality of use of the interactive systems.

2.2 Tools for working with guidelines

Many studies in the literature report tools for working with guidelines. These tools can be classified in two main categories: tools for automating user interface inspection and tools for managing guidelines sources. In the first category we will find tools such as EvalIris [3] that encode guidelines into algorithms for inspecting the user interface automatically. Very often, tools supporting inspection are used with Web applications where the code source (HTML/CSS) can be easily parsed [23][24].

Tools such as SIERRA [26], SHERLOCK [27] and GUIDE [30] are some examples of tools dedicated to the collection and maintenance of guidelines sets. These tools are able to manage usability guidelines in a more or less sophisticated way and permits to use in the design or evaluation of the specific interactive system. Most of these tools support the organization of several guidelines sources by adding specific metadata that facilitate searching of guidelines according to multiple criteria (ex. desired properties such as usability, application domain, etc.). Nonetheless, the presentation of guidelines remains a textual description.

A very few tools provide features for the management of user interface guidelines and the automated inspection of the user interface. In this category there is DESTINE [3] which can perform the assessment of guidelines formally expressed in a XML-compliant specification language called Guideline Definition Language (GDL). DESTINE includes a database for recording guidelines from diverse sources, however, not all guidelines can used for automated inspection.

2.3 Management of potentially conflicting guidelines

Several works report [11][28] problems associated to the management of guidelines sets. Vanderdonckt [5] discusses the potential occurrence of conflicting problems when selecting guidelines from diverse sources, for that those authors propose a dedicated process for selecting the best set of guidelines for a specific interactive system. Vogt [11] extends that work by proposing taxonomy of 11 types of problems associated to conflicting guidelines. Abascal et al. [28] explicitly mention that a step for the resolution of conflicting guidelines should be performed when selecting guidelines in an educational context (i.e. teaching guidelines); nonetheless they do not describe how conflict resolution can be specified. In [34] a set of unresolved problems are presented for the tools for working with guidelines. One of these specified problems is the maintenance of the guidelines. And the authors points to the conflicting guidelines as an examples of unresolved problem [29]. Finally, the state of the art about the process to get the most adequate set of guidelines is done and the most highlighted research in the definition of a process to get guidelines is [5]. In this research, a process to develop a set of guidelines is described using as the main point five milestones to get a tool for working with guidelines. Another work is [38] where defines a generic framework for the collaborative development of guidelines and standards involving many experts in the usage of guidelines.

To sum up, despite the fact that several works agree on the existence of potential conflicting user interface guidelines, there is not any research so far proposing a methodological approach for dealing with such conflicts. Existing tools for working with guidelines can handle diverse guidelines sources but they are not able to exhibit if conflicts between guidelines. Moreover, even if designers are able to solve the conflicts between guidelines, they have no support to document their arguments leading to the solution, which can be lost in future projects.

2.4 Traceability of conflicting guidelines

As far as we know, there is no work in the literature describing how systematically dealt with such conflicts. The resolution of conflicting guidelines requires the

systematic exploration of design options. In previous work [32][33] we propose the Design Rationale TEAM (Traceability, Exploration and Analysis Method) and the support tool called DREAM (Design Rationale Environment for Argumentation and Modeling) to support the systematic exploration of design options during the development process of interactive systems. Hereafter we illustrate the main concepts of the TEAM notation for describing guidelines. The combination of guidelines and the resolution of potential conflicts are described in section 4.

2.4.1 The TEAM notation and tool support

TEAM notation is an extension of MacLean and al.'s QOC (Question Option Criteria) [32] which allows the description of available options for a design question and the selection of an option according to a list of criteria. The TEAM notation extends QOC to record the information produced during design meetings, including:

- Questions that have been raised,
- Design options that have been investigated and the ones that have been selected,
- Criteria that have been used for evaluating the options considered,
- Requirements for the system and how they are supported by design options,
- Factors that have been taken into account and how they relate to criteria,
- Arguments and documents used to explain the design options,
- Task models corresponding to options,
- Scenarios that are used to compute, for each option the value of the criteria.

Fig. 1 shows a simple TEAM model that contains all elements require to describe guidelines. In the example below, the requirement for the web site "*provide access to data"* is represented by a square. The question raised during the web site design (represented by a square with rounded-corners) indicates two possible design options (represented by circles) to grant users with access to a Web site: "*provide direct access"* and/or "*ask* first *for login and password"*. The measurable criteria associated to design options are presented by isosceles triangles. The clip-shaped icon next to the item "*reach record in less than 20s*" links this criterion to the arguments and documents that can be used to measure it. The criteria can be directly connected to factors and sub-factors (represented by equilateral triangles) such in the case of the factor *security* and the sub-factors *efficiency* and *effectiveness* that are connected to the factor *usability*. The different types of lines between the criteria and options represent the fact that a given option can support (favour) a criterion (the line is bold) or not support it (the line is dotted). Thus, the option "*provide direct access*" supports *effectiveness* and *efficiency* but it does not support user data protection. The option "*ask for login and password*" strongly supports user data protection (bold line) but has an impact on *effectiveness* and *efficiency* (thin lines). TEAM supports more precise connection between elements (including absolute and comparative values) but this is not presented here due to space constraints.

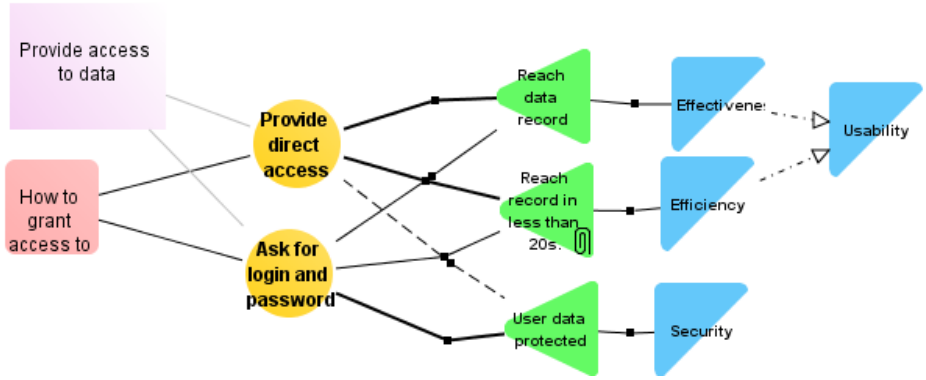

Fig. 1. Simple model showing the main elements supported by the notation TEAM.

TEAM models can leverage the design rationale process by helping designers to document their decisions and choices with respect to the many options available. Moreover, TEAM models can also help to decide to reuse (or not) design choices when facing an already experienced issue. TEAM notation is supported by the tool DREAM which supports the edition, recording and analysis TEAM diagrams [33].

2.4.2 Mapping individual guidelines to design rationale elements

Guidelines sources often provide information that can easily matched to factors and criteria that can be used to measure factors as illustrated in Fig. 2. Hereafter an example based on the *"WCAG guideline 1.1 text alternatives" is provided*, as bellow:

Description of the guideline: *"Guideline 1.1 Text Alternatives: Provide text alternatives for any non-text content so that it can be changed into other forms people need, such as large print, braille, speech, symbols or simpler language".*
Source: (WCAG) 2.0 (see [9])
Factor: *Accessibility*
Criteria: *Perceptibility*

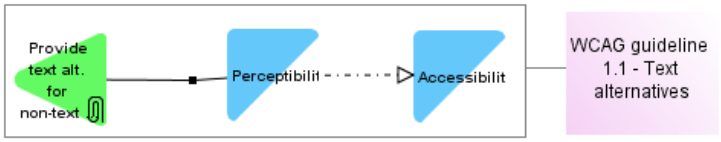

Fig. 2. Representation of individual guidelines using the notation TEAM.

It is noteworthy that the overall description of the guidelines is mapped to a non-functional requirement (represented by a square). Moreover, the guideline is connected through a box embedding the criteria "*provide text alt. for non-text*", the factor *accessibility* and the sub-factor *perceptibility*; such as composition shows that all these elements are part of the guideline description. The clip in the diagram indicates that there are additional documents explaining how the criteria can be assessed.

3 A Process for the Management of Guidelines Selection

Our approach assumes that design rationale methods can help in two ways: i) to select guidelines that can help to decide on the design options; and ii) to support the decision-making processing leading to the resolution of conflicting guidelines. Such hypotheses imply that the description of guidelines can be extended to represent design rationale elements.

3.1 Overview of process

Fig. 3 presents our approach for dealing with guidelines management. It encompasses three phases: the *organization of the guidelines* which is concerned by how relevant sources of guidelines are identified (step ❶), how guidelines are collected (step ❷), systematically described (step ❸) using a design rationale notation (R) and then stored into a database included in a database; the second phase describes how designers search into the database (step ❹) for suitable guidelines, which requires the selection of a guidelines subset that fulfil the needs of guidance for a given project (step ❺). At this point conflicting guidelines are detected and solved (step ❻); conflict resolution is then recorded CR into the database; the final phase describes how guidelines subsets can be effectively used (step ❼) during the development process as help for the design and/or evaluation of user interfaces.

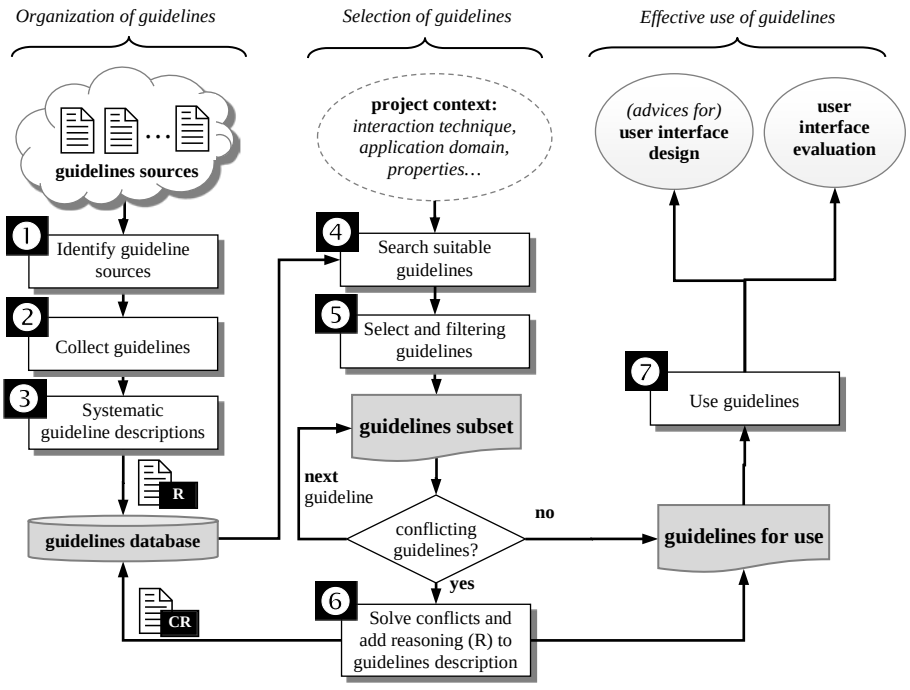

Fig. 3. Overview of a process for rationalizing the management of user interface guidelines.

Although the process can run independently, it is (partially) included in the Open-HEREDEUX project [13]. Open-HEREDEUX (short name of Open HEuristic REsource for Designing and Evaluating user experience) considers the wide concept of the user experience and includes four components: the Open Repository which stores all the needed information, the Adviser who lists the most adequate guidelines for a specific interactive system, the Scorer who helps to carry out the evaluation, and finally, the Results Analyzer where quantitative and qualitative results can be achieved. Currently, three tools developed in the context of the Open-HEREDEUX project support (partially) this process: the *open repository* tool that stores the set of guidelines, the Adviser which implement searching and filtering functions allowing to browse the open repository an select a sub set of guidelines (steps ❹ and ❺), and the tool *Scorer* which is dedicated to presented previously selected guidelines during the evaluation phases (step ❼).

3.2 Systematic guidelines description

In order to provide a seamlessly association of user interface guidelines, design rationale process and guidelines storage an appropriate schema for representing guidelines in a database is needed. It is worthy of notice that guidelines descriptions can include diverse attributes as discussed in [33]. However in **Table 1** just a small subset of attributes that are relevant for understanding of the contribution aimed for this paper is provided. This information is stored in the guideline database (handled by the Open Repository tool) to facilitate the later detection of conflicts and the extraction of results.

Table 1. Description of the database attributes for describing guideline.

Attribute	Description	Representation in the database
Guidelines description	Statement describing the guideline	text
Source	Source of the guidelines	text, pointers to documents
Application domain(s)	Known applications domain where the guideline is used	text
User interface components	User interface that can affected by the guideline (e.g. text, images)	text
Keywords	Metadata and keywords that can be improve searching	text
Factors	List of factors and sub-factors covered by the guidelines	TEAM model
Criteria	List of criteria that should be measured	
Importance of factors	A value that tells the impact of a particular factor to the guideline	Likert scale (1-5)
Potential conflicting guidelines	List of known conflicts and the solutions	pointers to other guidelines and TEAM model resolution of conflicts

TEAM models might occur in the guidelines database twice: i) for describing individual guidelines (in combination with factors and criteria as shown in section 2.4.2); and ii) for describing how *potential conflicting guidelines* have been solved (as shown in section 3.4). In order to help to decide between recommendations proposed by potentially conflicting guidelines, it is import to know each guideline is related to

factors. For this purpose, the proposed schema also allows to define the *importance of factors* in a given guidelines in Likert scale of 1 to 5. **Table** 2 provides a view at glance on what could be the impact of factors/sub-factors with respect to the guideline "*WCAG guideline 1.1 text alternatives*". As we shall see, this guideline is considered to have little impact on usability sub-factors and as far as accessibility is a concern, the perceptibility for blind and blind-deaf is more important that just for perceptibility for deaf. However, this guideline is not addressing security factors, which remain blank.

Table 2. Weights of factors for the guideline "*WCAG guideline 1.1 text alternatives*".

		lowest	←	priority	→	*highest*
Factor	***Sub-Factor***	1	2	3	4	5
Usability	Effectiveness	x				
	Efficiency	x				
	Satisfaction	x				
Accessibility	Perceptibility (for blind)					x
	Perceptibility (for deaf)	x				
	Perceptibility (for blind-deaf)				x	
	Perceptibility (cognitive disabilities)			x		
Security	User data protection					

3.3 Select and filtering guidelines

Ideally, guidelines databases should contain exhaustive set of guidelines covering diverse sources and application domains. Nonetheless, huge entries of guidelines are difficult to handle. So quite it is often useful to create a subset which contains only guidelines that are really relevant to the development of a given applications. In Fig. 3, the selection and filtering of guidelines is made in the step 5. In the Open-HEREDEUX approach this is done be the tool called Adviser will pre-select appropriate guidelines for a specific interactive system by taking into account the functionalities of the system, its features and its components.

The sets of guidelines could include guidelines by different authors, for different users, for different factors, for different applications domains, etc. All these elements can be used as inputs for searching and selecting guidelines. So that our approach not only supports searching but it also allows the definition of weight that factors might have for a project. For that designers must select the list of factors and then determine the priority of each factor and/or sub-factors. The priority for each factor can be defined using a form that is build using a Likert scale similar to that presented in **Table 2**. For example, a designer can choose that the security is the most important property for a specific system or that the accessibility for blind people is the priority. It also possible to remove factors and/or sub-factors that are not relevant for the project, for example one can decide that perceptibility for deaf is not going to be taken into account. In a first moment, the selection of factors and sub-factors will help to retrieve the list of entries in the databases and then select only those guidelines that treat those factors. As we shall see latter on, the weight given to factor will help to decide the resolution of conflicting guidelines when then occurs. Once a sub-set of guidelines has been selected, then it is time to check for potential conflicts, as described below.

3.4 Conflict management and guidelines cleaning

The combination of different guidelines sources into a single database will eventually end-up with entries that might contain similar statements employing different terms (which might be confusing), duplicated entries, guidelines that refer to design elements that are not relevant to the project, and potentially guidelines which are strongly conflicting i.e. proposing contradictory recommendations. In all these cases a cleaning process is required together with a resolution process in case of strong conflicts.

In our approach, the guideline database is supposed to contain all necessary information for helping designers to detect conflicts between guidelines. However, (as listed above) not all conflicts look alike and many types of connections between guidelines might occur. For example, considering the existence of two guidelines (**G1** and **G2**) the following scenarios and types of connections can be defined:

- Equal (E): guidelines can be considered very similar or equal.
  Ex. The website guideline G1: "Is user provided with the essential information to carry out each task?" is the same as G2: "Only show essential information" for mobiles.
- More general (MG): G1 is more general than G2.
  Ex. The website guideline G1: "Are the same elements grouped and located in the same place?" is more general than G2: "When designing an application, optimize edit view for data entry, grouping related items and prioritizing more commonly edited items at the top of the screen" for mobiles.
- More specific (MS): G1 is more specific than G2.
  Ex. The guideline G1: "Are the required values always marked using the same method?" is more specific than G2: "Required fields are marked".
- Conflict (C): There is a clear contradiction between both guidelines.
  Ex. The website guideline G1: "Is there in the top and the bottom of the page information about where are the users and the last page visited?" is conflicted with G2: "Do not repeat the navigation on every page" for mobiles.
- Superseded (S): One guideline presents a superseded of the other one.
  Ex. The guideline G1: "Use Audio CAPTCHA to prevent spam" is a superseded version of the guideline G2: "Use graphic CAPTCHA to prevent spam."

In all the scenarios above, the set of guidelines should be cleaned-up before use. The cleaning is part of the selection and filtering process described in the step 5 of **Fig. 3**. If the conflict is of type E, the solution could be use either G1 or G2, without distinctions but one of them should be removed to avoid redundancy in the subset. If the conflict is an MG, the solution is to choose G2 and then to remove G1 from the subset of selected guidelines. In this case, it is better to use the most specific one so, the guideline "*When designing an application, optimize edit view for data entry, grouping related items and prioritizing more commonly edited items at the top of the screen*" is chosen. If the conflict is of type MS, the solution is to choose G1. Otherwise, the selected guideline will be "*Are the required values always marked using the same method?*" The scenarios (E), (MG) and (MS) are relatively simple to detect and to treat. However, when a guideline is superseded (S) by another or it is in

conflict (C) with other guideline, further analysis is required. The goal of our approach is to help designers to specify systematically arguments and decisions.

In our approach, the first step to solve the conflict is to align the two (or more) guidelines. The way to get the design rationale is to construct a TEAM diagram from the individual TEAM diagrams that have be stored in the database as part of the guideline definition. The element question in the TEAM notation should be provided by the designer as it is a factor dependent of the context of the project. The options can be provided either but the recommendations in the guidelines description or manually provided by the designers. Finally, the weights associated to factors and each guideline is depicted in the diagram. The rest of the resolutions of the conflict are done by the designers with the help of the tool DREAM. When a solution is found, a diagram containing the solution is recorded. Moreover, every guideline considered 'in conflict" is tagged in the database. So in the future it would be possible to retrieve the solutions found in previous projects.

When a solution is finally found, the appropriate guideline (G1 or G2) becomes part of the subset of guidelines going to be used in the project (**Fig. 3**, step 7). To sum up, the main aim of the first stages to manage clashes is to save all the needed information about the whole sets of guidelines, the interactive system and the context of use. If this information is saved properly, the next steps will be done and will be easier to get. The next section illustrate the whole proceed in a concrete case study.

4 Case Study

The example provided in this section is issued from the application of our approach in the context of the Ubiloop project. Hereafter we only illustrate a few guidelines addressing the design of *captchas* which, as we shall see, are obviously conflicting. This example was deliberated chosen for focusing the discussing on the process of selecting and describing conflicts between the guidelines rather than improve the knowledge of conflicts of this particular element of the design. Due to space reasons, we don't describe the preliminary steps required for creating the database of guidelines. We assume that guidelines are already stored into a database but there is conflicting guidelines have not been reported (yet).

4.1 Informal description of the case study

The Ubiloop is concerned by the development of solutions for improving the quality of the environment of the city using mobile and information technologies. The approach proposed by Ubiloop is to offer an incident reporting systems that allows citizens to report incidents in their neighbourhood that might affect the quality of life, such a potholes, broken street lamps, graffiti, etc. The requirements for this application include the use Web technology. Moreover, the application should run in whatever platform/or devices citizens might have at their disposal, which might include smartphones. In our working scenario, mobile technology is an essential ingredient because it allows users to make a report just after problems have been detected when all the details about the incidents are still fresh in users' mind.

We also have identified some factors that are important for this kind of application including the usability, as everyone should be able to use the application and perform a report in a minimal time. Accessibility is an important factor enhanced by regulations. Security becomes an important factor as the kind of application we have in mind can suffer attacks from spambots tools that can shutdown Web servers with massive spams and/or reduce the trust on the information collected.

Based on these requirements, we have searched the guidelines database for the three applications domains that are concerned by our project: Web applications, mobile and incident reporting. The selection process is described below.

4.2 Selection of guidelines

There is huge set of reference in the guideline database that can provide suitable recommendation for dealing with the design of application domains. A first search reveals as many as 117 entries for guidelines including: 82 guidelines for Websites, 84 guidelines for mobile phones and 11 guidelines for incident reporting forms.

A first analysis of these 177 entries reveals several overlaps and conflicts between guidelines. For example, there are 16 clashes between guidelines for building Web applications and guidelines for building incident reporting systems. As many as 138 conflicts concern guidelines for the development of Web and mobile applications. Finally, 19 clashes are detected between guidelines for incident reporting forms and guidelines for mobile applications. These numbers are better presented in **Fig. 4**.

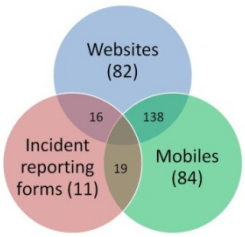

Fig. 4. Overlapping of guidelines issues from three guidelines sets: Web sites, Incident reporting systems and mobile applications.

Therefore, designers should compare if it is possible to apply all these guidelines or, in a contrary way, they detect clashes and they have to choose one or other. So, the next step was to clean the list of guidelines by removing duplicated entries.

4.3 Management of guidelines related to captchas

In our scenario, designers' goal is to develop an interactive system and for that they need advices which in our case are provided by guidelines. The first question designers have to answer is how users will have access to the data on the Web site so that they will be able complete the incident report form. The user should insert all the information of the incident and the system should assure the security of the transmission and the privacy of the personal information. In addition, the interface should be able to control that only real users fill in the form. It means that the

interface should present the needed security restrictions to prevent spam or robot messages.

A possible solution to the questions above is to introduce in the user interface a new component to cover the requirement of the security: a captcha. A captcha stands for "Completely Automated Public Turing test to tell Computers and Humans Apart"[1]; it is a type of challenge-response test used in computing as an attempt to ensure that the response is generated by a person. Captchas help in ensuring that all reports have been inserted by citizens and not by bots because humans can read distorted text and or sound but current computer programs can't. By asking users to fill in a form field the letter shown as distorted text/sound, it is possible to infer that the other fields in a form were duly completed by a human. Several possible implementations of captchas exist. **Fig. 5** shows some examples of design options for captchas including visual, audio and visual-audio captchas.

a) visual captcha *b) audio captcha* *c) visual-audio captcha*

Fig. 5. Examples of design options for captchas.

Whilst captchas help to ensure security, they can reduce usability and accessibility as pointed by the guidelines in **Table 3**.

Table 3. Main conflicts between guidelines related to Captcha.

Design Questions	**G1(Captcha guidelines)** Sources: [35] [36]	**Type of conflict**	**G2(Website guidelines)** Sources: [12][36]
	Visual Captcha		
How do we can access to the complete meaning of the graphic?	Do not add alternative text for protecting user data.	C	Add alternative text to images
How is the text presented in graphic captcha?	Present distorted text to difficult the text recognition.	C	Present text clearly.
	Audio Captcha		
How do we can access to the complete meaning of the audio?	Do not add any alternative information	C	Add Braille information.
How is the information presented in audio captcha?	Noisy should be added deliberately.	C	Use clear messages
How can users understand the language of the captcha?	The audio should speak in a specific language	C	The multi-language option appears
	Text Captcha		
Is the text easy to read?	Use easy and understandable questions	MG	Text can be resized inside the browser.

[1] Further details at : http://www.captcha.net/

One of the most striking conflicts can be translated as follows:

- G1: Prevent spams from bots.
- G2: Provide text alternative for non-textual elements.
- Type of clash (G1 in respect to G2): Conflict (C)
- Rational for describing the conflict: as providing a text alternative for non-visual element captcha (as it is done with the attribute *alt* for images) will remove the security protection as programs can also read the *alt* attribute from HTML pages.
- Question: How is the text presented in the graphic captcha?
- Rational for deciding the conflict: importance of security versus accessibility and usability should help to decide if we keep (or not) captcha as a design option, and is so, which is the most appropriate design option for implementing captcha (for this is necessary to describe with which kind of accessible issues we are dealing with: visual impairments, audio, etc.).

The solutions for these conflicting guidelines requires a deep analysis of the associated trade-offs and the weight given to each factor can help designers to make a decision. For example, as security is very important in our case study, captchas should be implemented even if they can reduce the usability of the final user interface. If accessibility for blind people is important, then should go for a visual-audio captcha instead of a simpler visual captcha. It is important to note that whatever is the decision, it will represent an infringement of some guidelines. In order ensure that the decision explicitly represents trade-offs, a design rationale is necessary. When conflicts between guidelines occur, designers resolve it and document the solution. For that our approach proposes the use of design rationale. In **Fig. 6** how designers can create a TEAM model for help the decision-making process leading to the best design options for using captchas in the context of the Ubiloop project is illustrated.

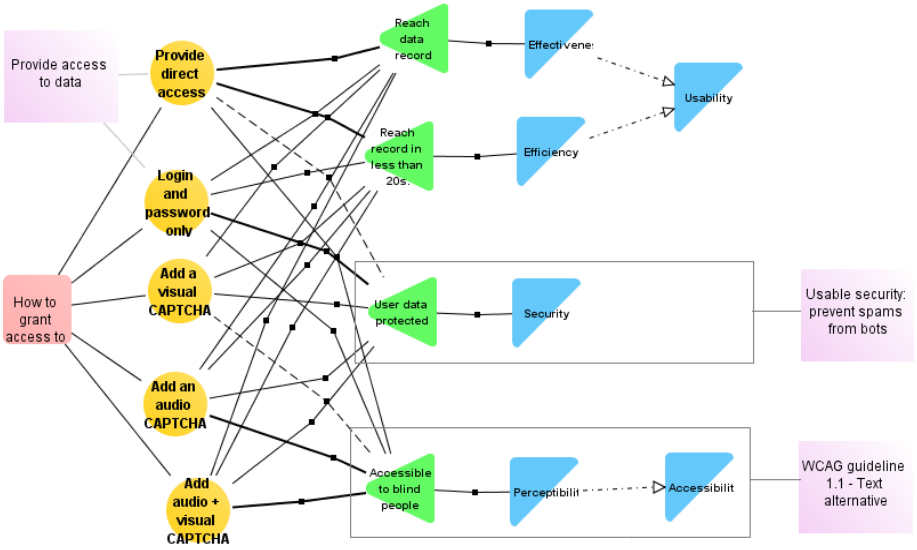

Fig. 6. TEAM model describing the rationale for analysing trade-off between captchas with respect to the requirements of the Ubiloop projects and potentially conflicting guidelines.

The constructions of such TEAM models must be done by designers with the help of the tool DREAM. However, assuming that we already have in our guideline database the design rationale used to describe a set of individual guidelines using the TEAM notation (see section 2.4.2), it is possible to automatically retrieve these individual models and insert them automatically in that TEAM model used to analyse the trade-offs and conflicts between guidelines. This situation is illustrated in **Fig. 6** by the two guidelines "*WCAG guideline 1.1- text alternative*" [9] and "*Usable security: prevent spams from bots*" [9][35].

As we see in **Fig. 6**, the TEAM notation supports the observation of the relationships between guidelines and factors, thanks to its simplicity and readability, intended to be understandable by most of the actors involved during design (graphic designers, developers, customers,…),. The weights (visible through the connecting lines) suggest that the option *provide direct access* to the application does not comply with the security (dashed line in the connection) so this solution should be selected. The many alternative implementations of captchas do favour security but only visual and audio-visual captchas are accessible for blind users, so that one on these options can be selected. This diagram explicitly shows the compromises that have been made between the guidelines for deciding the final solution to the problem. Once a decision was made for the Ubiloop project, it is recorded into our guidelines database so that it will become available for the next time when designers are confronted to a similar problem. The solution of this conflict also becomes accessible when browsing the guidelines in the database, so that it would be possible to explore trade-offs and known conflict with other guidelines.

Conflicting guidelines can be perceived at a glimpse at TEAM diagrams by looking at the lines connecting guidelines and design options. Our claim on "easier to observe" is based on the fact that a glimpse at a TEAM diagram allows to detect the divergent 'favouring lines' when some paragraph of text to would be required to explain the same thing. When diagrams are getting larger (in case of multiple conflicts only) visualization techniques such as bi-focal browser have been proposed [39] as well as colour matrixes [33].

5 Discussion

In this paper some of most striking questions related to selection and management of conflicting guidelines have been revised. Moreover, we have shown how a design rationale approach can help for dealing with trade-offs and design choices associated to guidelines. The same approach can also be used to trace the rationale behind the evolution of guidelines recommendations that are updated to reflect changes in either user behaviour or improvements of the technology.

Whilst the existence of potentially conflicting guidelines is almost common sense, as far as we know these problems are barely documented in the HCI literature. Nowadays the problems of conflicting guidelines become more important because new applications are being created using a combination of technologies that were before known in specific application domains; as we have illustrated in our case study for the Ubiloop project.

The main goal of the present work is help users of large collections of guidelines (in particular designers, developers and evaluators of interactive systems) to deal with trade-offs between conflicting guidelines. The present work puts at light these problems related to the management of conflicting guidelines. On one hand we hope to deepen the knowledge on the management of guidelines. On the other hand we expect to help designers to understand the uses and misuses of guidelines. Indeed, guidelines are world-wide used for providing guidance to the projects but there is little evidence on how designers solve conflicting recommendations.

The approach presented in this paper is a possible solution but it imposes some constraints for the description of guidelines according to the TEAM notation. Nonetheless, it is not necessary to have all the guidelines systematically described to get the benefits. The description of guidelines can be done incrementally and the database can contain entries that are not represented using the TEAM notation. Moreover, we suggest that conflicting guidelines should be represented and documented only when they occurs in real projects for two main reasons: first of all because we need the contextual elements given by the real project to decide the best option; secondly because the modelling activity can be time-consuming so it is better to make an effort when we can have an immediate benefit.

It is noteworthy that our approach for describing guidelines can be triggered either if we need guidelines for the design and/or evaluation of interactive systems. However, in some cases, the solution of conflicting guidelines will be achieved after several iterations in the development process of the application. For example, if we are in a design phase, we can select the guidelines, we can provide a TEAM model to complete the description of individual guideline and, possible detect that guideline might be in conflict with other guidelines. However, it might happen that the resolution of the conflict could not be done at design time as it might require some user testing to decide the trades-offs and the arguments allowing to solve the conflict. Thus only when evaluations have been performed we can go back and record the solutions for the conflicts identified previously. Whilst the decision making process leading to the resolution of the conflicting guidelines is done by designers in an ad hoc manner, the description of the solution should be systematic and exhaustive. In this paper we advocate for a systematic approach to guidelines conflict management using a supporting method and framework for the reuse of solutions to common recurrent conflicts.

This work also brings some questions that might influence the development of new tools for working with guidelines. Indeed most of the tools for working with guidelines can be described as a simple catalogue or database with searching facilities [10]. In fact there is little room in these tools for an active support of the decision-making process that occurs when building new applications. Indeed, our approach cannot only benefits designers with advices for the user interface but also call them for providing feedback about their decisions.

6 Future work

Currently we are working to fully implement our approach in the Open-HEREDEUX project [13] that includes, among other components, the Open Repository, the Adviser

and the Scorer. So, all the information about the conflicts will be stored in the Open database of guidelines. Moreover, the Adviser would be the processor who will detect the conflicts and, using the design rationale notation, who will send the solutions to the Open Repository to save them. Up to now, the Open Repository, the Adviser and the Scorer are available to get the most suitable set of guidelines for designers and evaluators. The next step is the implementation of the Result Analyzer and the methodology presented here to detect conflict between guidelines.

This paper illustrates the feasibility of our approach; however it has only been applied in a single real-world project. Further studies are necessary to investigate the potential of adoption of our approach by a larger community of designers and developers of interactive systems. With this respect, we want to explore in the future the participation of online communities that could help by creating individual TEAM models for guidelines and reporting the potential conflicting guidelines. The motivation of providing these feedbacks is that one can also get benefits of information provided by other members of the community. That is an alternative for creating a collective intelligence around the management of conflicting guidelines. In such context, design rationale because even more important because it can help the contributors to provide the right arguments for solving the conflicts. As the trades-off and design options are systematically exhibited by TEAM models, it might reduce the ambiguities associated with the interpretations of comments send by contributors.

The approach is aimed at recording the solutions for conflict resolutions so that they become available for future use. Thus a database is enriched with knowledge about the conflicts. Such information can be exploited in the future for understanding how conflicts have been solved. We can also envisage using such information with recommender systems that could provide real time advices when selecting guidelines sources, for example.

Future work will include usability and scalability studies based on empirical studies with designers and larger cases studies employing a larger set of guidelines.

Acknowledgements

The work has been supported by Universitat de Lleida for pre-doctoral fellowship to Llúcia Masip and the EU project FEDER Ubiloop.

References

1. Tidwell, J. Designing interfaces, Sebastopol, California, USA: O'Reilly, 2005.
2. van Welie, M., van der Veer, M., 2003. Pattern languages in Interaction Design: Structure and Organization. Proc. of INTERACT 2003. IOS Press, IFIP, 2003, pp. 527-534.
3. Abascal, J., Arrue, M., Fajardo, I., Garay, N., and Tomás, J. 2004. The use of guidelines to automatically verify Web accessibility. Univers. Access Inf. Soc. 3, 1 (Mar. 2004), 71-79.
4. Winckler, M., Bernhaupt, R., Pontico, F. (2010) Challenges for the Development of User Interface Pattern Languages: A Case Study on the e-Government Domain. Int. Journal on WWW/INTERNET, IADIS Digital Library, Vol. Vol. 8 N. 2, Dec. 2010. p. 59-84.
5. Vanderdonckt, J. Development milestones towards a tool for working with guidelines. Interacting with Computers 12(2): 81-118 (1999)
6. Van Duyne, D. K., Landay, J. A., and Hong, J. A., 2002. The Design of Sites: Patterns, Principles, and Processes for Crafting a Customer-Centered Web Experience. Reading, MA: Addison-Wesley, 2002.

7. Grill,T. and Blauhut, M., 2008. Design Patterns Applied in a User Interface Design (UID) Process for Safety Critical Environments (SCEs). 4th Symposium of the Workgroup Human-Computer Interaction and Usability Engineering of the Austrian Computer Society, Graz, Austria, pp. 459-474.
8. Kinert, T., 2009. User-Centered Interaction Design Patterns for Interactive Digital Television Applications. Springer London. 315 p. ISBN 978-1-84882-274-0.
9. W3C (2008). Web Content Accessibility Guidelines 2.0. W3C Candidate Recommendation April 2008.[http://www.w3.org/TR/WCAG20/]
10. Vanderdonckt, J., Farenc, C. (Eds.) Tools for Working with Guidelines. Springer-Verlag, London. (2001)
11. Vogt, T. Difficulties in Using Style Guides for Designing User Interfaces. Tools for Working with Guidelines. J. Vanderdonckt, C. Farenc (Eds.), Springer-Verlag, London, pp. 197-208.(2001).
12. Martin, D., Rouncefield, M., and Sommerville, I., 2002. Applying Patterns of Cooperative Interaction to works (Re)Design: E-Government and Planning. In Proceedings of ACM CHI 2002, April 20-25, 2002, Minneapolis, USA. Pages 235-242.
13. Masip, L., Oliva, M., Granollers, T. OPEN-HEREDEUX: open heuristic resource for designing and evaluating user experience. In Proc. of INTERACT'2011. Springer-Verlag LNCS 6949, Berlin, Heidelberg, pp. 418-421. (2011)
14. Pontico, F., Winckler, M., Limbourg, Q. Organizing user interface patterns for e-Government applications, Engineering Interactive Systems (EIS), Salamanca, Spain, March 22-24, 2007. pages 601-619. ISBN:978-3-540-92697-9.
15. Abdo Beirekdar, Marc Keita, Monique Noirhomme-Fraiture, Frédéric Randolet, Jean Vanderdonckt, Céline Mariage: Flexible Reporting for Automated Usability and Accessibility Evaluation of Web Sites. INTERACT 2005: 281-294
16. ISO. International Standard. ISO 13407. Human-centered design, processes for interactive systems (1991).
17. Prates, R., de Souza, C., Simone D. J. Methods and tools: a method for evaluating the communicability of user interfaces. interactions pp 31-38. (2000)
18. Jiang, O. De Bruijn, O.; A. De Angeli. “The Perception of Cultural Differences” in Online Selfpresentation. T. Gross (Eds.):INTERACT 2009, Part I, LNCS 5726, pp. 672–685.
19. Thevenin, D., Coutaz, J., Plasticity of User Interfaces: Framework and Research Agenda. INTERACT’99. Conference on Human-Computer Interaction, Vol. 1, pp. 110-117. (1999)
20. González, J.L.; Padilla, N. and Gutiérrez F. From Usability to Playability: Introduction to Player-Centred Video Game Development Process. Proceedings of the 1st HCD 09, Masaaki Kurosu (Ed.). Springer-Verlag, Berlin, Heidelberg, pp 65-74. (2009)
21. Avizienis, A., Laprie, J.C., Randell, B., Landwehr, C. Basic Concepts and Taxonomy of Dependable and Secure Computing. IEEE Trans. Dependable Secur. Comput. 1, 1, (2004)
22. Blackwell, A. F., and Fincher, S., 2010, PUX: patterns of user experience. Interactions, vol. 17, no. 2, pp. 27–31.
23. Xiong, J., Winckler, M. (2008) An investigation of tool support for accessibility assessment throughout the development process of Web sites. Journal of Web Engineering (JWE) (special issue about Web Usability and Accessibility). Rinton Press. Vol.7 No.4. pages: 281-298.
24. Xiong, J., Farenc, C., Winckler, M. Analyzing Tool Support for Inspecting Accessibility Guidelines during the Development Process of Web Sites. In Proc. of IWWUA’2007. Nancy, France, December 3-7, 2007. Springer LNCS 4832. pages 470-480.
25. http://www.evengrounds.com/es/node/120
26. Vanderdonckt, J. Accessing guidelines information with SIERRA. In Proc. 5th IFIP TC 13 INTERACT’95, Chapman & Hall, London. 1995. pp. 311-316.
27. Grammenos D., Akoumianakis D. and Stephanidis C. Integrated support for working with guidelines: the Sherlock guideline management system. Interacting with Computers.

2000. N3, Vol. 12. pp. 281-311.
28. Abascal, J.,Nicolle, C.The application of USERfit methodology to teach usability guidelines.Tools for Working with Guidelines. J. Vanderdonckt, C. Farenc (Eds.), Springer-Verlag, London pp 209-216 (2001).
29. J. Abascal, C. Nicolle: Why Inclusive Design Guidelines. In C. Nicolle, J. Abascal (Eds.): Inclusive Design Guidelines for HCI. Taylor & Francis. London. 2001. pp. 3-13.
30. Henninger S. A methodology and tools for applying context-specific usability guidelines to interface design. Interacting with Computers. Elsevier, 2000. N3/Vol. 12, pp. 225-243.
31. Partarakis, N., Mourouzis, A., Doulgeraki, C., Stephanidis, C. A portal-based tool for developing, delivering and working with guidelines. In Proc. of UAHCI'07. Springer-Verlag, Berlin, Heidelberg, 507-516.(2007)
32. Lacaze, X., Palanque, P., Barboni, E., Bastide, R., Navarre, D., From DREAM to Reality : Specificities of Interactive Systems Development With Respect to Rationale Management. In Dutoit, A.H., McCall, R., Mistrík, I., Paech, B. (eds.) Rationale Management in Software Engineering: Concepts and Techniques, Rationale Management in Software Engineering. Springer Verlag 2006, pp.155-170.
33. Martinie, C., Palanque, P., Winckler, M., Conversy, S. DREAMER: a Design Rationale Environment for Argumentation, Modeling and Engineering Requirements. In proceedings of the 28th ACM International Conference on Design of Communication (SIGDOC'2010), September 26-29, 2010, São Carlos, Brazil. ACM Press. pp. 73-80.
34. Cranor, L.F., and Garfinkel, S. (eds.) Security and Usability: Designing Secure Systems that People Can Use. O'Reilly Media, 744 pages. ISBN-10: 0596008279
35. http://www.evengrounds.com/es/node/120
36. http://www.evengrounds.com/es/node/584
37. Mariage, M., Vanderdonckt, J., Pribeanu, State of the Art of Web Usability Guidelines. In: The Handbook of Human Factors in Web Design (1999).Publisher: Lawrence Erlbaum Associates, Pages: 688-700. ISBN: 080584612. DOI: 10.1.1.58.4494
38. Dearden, A., and Finlay, J. 2006. Pattern languages in HCI: A Critical Review. Journal of Human-Computer Interaction, Volume 21, Issue 1 March 2006, pp. 49 – 102.
39. Palanque P. & Lacaze X. DREAM-TEAM: A Tool and a Notation Supporting Exploration of Options and Traceability of Choices for Safety Critical Interactive Systems. In Proceedings of INTERACT 2007, LNCS, Springer Verlag, p. 230-245.