9 Interaction Semantics Consistency in Cross-Platform Design

Fabio Paternò, ISTI-CNR, Via G.Moruzzi 1, 56126 Pisa, Italy, fabio.paterno@isti.cnr.it

Giulio Mori, ISTI-CNR, Via G.Moruzzi 1, 56126 Pisa, Italy, giulio.mori@isti.cnr.it

In this position paper we discuss some issues related to continuity in cross-platform design, including when it addresses platforms supporting different modalities. We argue that one key aspect is to be able to support continuity in terms of interaction semantics for those tasks that are deemed meaningful in the target platforms. We also discuss a method for interface transformation from desktop to mobile platform following this approach

INTRODUCTION

Cross-platform design raises interesting issues. On the one hand there is a need for adaptation, and thence for change, to the features of the changing platform in order to take into account the available resources and exploit them at best for obtaining usable results. On the other hand, users always access the same application, albeit through different platforms, and thus they prefer to reuse the knowledge of the system interaction that they have acquired while accessing through one device also when they access through another.

Design of multi-device interfaces and migration in such environments has been one of the main research areas considered in my group over the last four-five years. We have been working both on authoring environment for designing multi-device interfaces [1] and on migratory interfaces and related environments [2]. The latter are interfaces that are able to dynamically migrate from one device to another and allow the user to continue their task from the point in which they left off in the source device.

To address such issues we have developed a XML-based language for conceptual user interface description based on the semantics of the user interactions and how designers aim to structure the user interface. Such elements are general and can be applied for any interactive device we want to address.

Consistency in user interfaces in general means that in similar situations the same design solution is adopted. Thus, there is logical coherence in applying design criteria. This can involve design at various levels (choice of the graphical attributes, interaction techniques, navigation structure, …). We consider a platform as a set of devices that share similar interaction resources (such as the desktop, the PDA, the vocal one). Thus, when we want access the same application through a different platform we have to consider that there is a substantial change in terms of available interaction resources and modalities and, consequently, we have to change something in the user interface.

TASKS AND CROSS-PLATFORM DESIGN

One issue is that one-fits-all does not work. This means that it is not meaningful to try to support all tasks for all platforms. The big push for cross-platform design comes with the advent of mobile devices, which people can use when they are on the move. This means that there is a clear distinction regarding what it is meaningful to do with a desktop and with a mobile device. For example, people can use a desktop to compare airfares and make reservations, while the mobile device can be used to check the real-time status of a particular flight. Likewise, the desktop is well-suited to reading a movie review and/or watching a trailer but not to purchasing a cinema ticket to avoid the line. There are many tasks that are not suitable at all for a mobile device. For example, currently in Europe with the advent of UMTS there is an interest in proposing football matches over phones. This seems senseless even with the last generation mobile phones, which have large displays and better connectivity. People like watching football matches but generally only when they are comfortable sitting and watching a large display in front of them. This allows them to appreciate how the footballers play, the details of the action, the tactics adopted and so on. Few people would ever do this through a mobile phone, even if it is technically possible, because of the small display and discomfort of using it on the move. Also considering that a football match lasts 90 minutes, this would be a terrible experience. What they could appreciate with a mobile device is a different task: receiving real-time updates regarding the match score. Another useful option is to have just the vocal description of the match that can be followed while driving the car or on the move. Thus, they could do this in parallel with other activities.

TERESA XML

Once we have identified the tasks that are meaningful to support for each platform then we have to identify how they performance can vary according to the target platform. The idea is to first identify potential interactors in terms of their semantics (how they can change the state of the application) and then to generate corresponding interfaces depending on the platform features. All devices belonging to a given platform will receive an interface with consistent implementation.

An abstract user interface is structured into presentations and connections indicating how it is possible to move from one presentation to another. Each presentation is structured into interactors and composition operators. We have defined a number of composition operators, which aim to capture communication effects that often designers want to achieve when they structure their user interfaces. The purpose of the composition operators is to indicate how to put together interactors. Each composition operator is associated with a communication goal. Depending on such goals, different implementation techniques will be used to support the composition operator. Figure 1 shows an example of a Web page taken from a frequently accessed Web site. We can note how the designer used various techniques to highlight groups of related interface elements. On the top there are elements that are ordered according to the potential user interest. Some elements are grouped using implementation techniques such as same background, same structure, bullets and so on. There are elements that are related to the rest of the Web site, such as the search element. Other elements are highlighted using large image and fonts because they are considered important.

In general, the composition operators can involve several interactors or even compositions of interactors. In addition, their definition is modality-independent. They are:

- Grouping (G): indicates a set of interface elements logically connected to each other;
- Relation (R): highlights a relation (usually one-to-many) among some elements; one element has some effects on a set of elements;
- Ordering (O): some kind of ordering among a set of elements can be highlighted;
- Hierarchy (H): different levels of importance can be defined among a set of elements.

There are different types of interaction elements depending on the type of task supported. We have selection elements (to select between a set of elements), edit (to edit an object), control (to trigger an event within the user interface, which can be useful to activate either a functionality or the transition to a new presentation). There are different types of only_output elements (text, object, description, feedback) depending on the type of output the application provides to the user: a textual one, an object, a description, or a feedback about a particular state of the user interface.

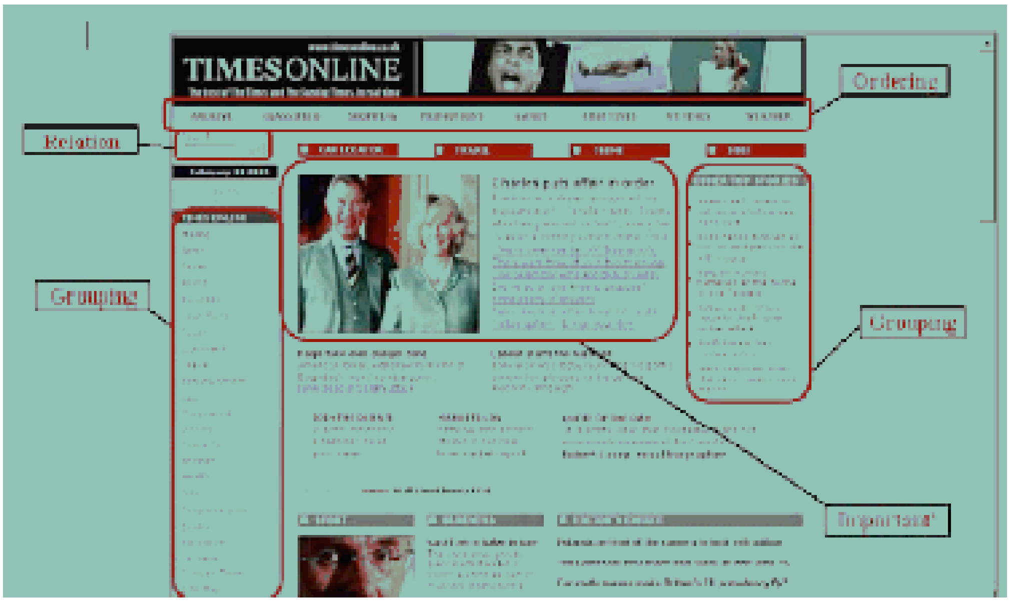


Figure 1: Web page with indication of some associated communication goals.

SEMANTIC PLATFORM REDESIGN

The problem of adapting the interface to different platforms can be addressed in many ways. Low-level syntactical transcoding or just resizing elements do not seem able to provide general solutions: they often generate poor results in terms of usability because they follow rigid rules and mainly try to fit the same design into different devices. We have developed a solution [3], which is based on platform-dependent semantic redesign. Semantic redesign means that transformation from one platform to another is based on the use of semantic information and not only on the analysis of the low-level implementation. In our case, such semantic information is contained in logical descriptions of the user interfaces that also capture the possible tasks users intend to accomplish.

In this section we discuss how our platform-dependent redesign transformation works considering the concrete /abstract description of the user interface. Given the limited resources in screen size of mobile devices, desktop presentations generally must be split into a number of different presentations for the mobile devices. The logical levels provide us with some semantic information that can be useful for identifying meaningful ways to split the desktop presentations along with the user interface state information (the actual implemented elements, such as labels, images, …). The redesign module analyses the input from the logical descriptions and generates an abstract and concrete description for the mobile device from which it is possible to automatically obtain the corresponding user interfaces. The redesign module also decides how abstract interactors and composition operators should be implemented in the target mobile platform.

In order to automatically redesign a desktop presentation for a mobile presentation, we need to consider semantic information and the limits of the available resources. If we only consider the physical limitations we may end up dividing large pages into small pages that are not meaningful. To avoid this, we also consider the composition operators indicated in the logical descriptions. To this end, our algorithm tries to maintain interactors that are composed through some operator at the conceptual level in the same page, thus preserving the designer’s communication goals. However, this is not always possible because of the limitations of the target platform. In this case, the algorithm aims (when possible) to equally distribute the interactors into mobile device presentations. In addition, splitting the pages requires a change in the navigation structure with the need for additional navigator interactors that allow access to the newly created pages. More specifically, the transformation follows these main criteria:

- The presentation split from desktop to mobile takes into account the composition operators because they indicate semantic relations among the elements that should be preserved in the resulting mobile interface. Another aspect considered is the number and cost of interactors. The cost is related to the interaction resources consumed, so it depends on pixels required, size of the fonts and similar aspects.

- The implementation of the logical interactor may change according to the interaction resources available in the target platform.
- The connections of the resulting interface should include the original ones and add those derived from the presentation split.
- The images should be resized according to the screen size of the target devices keeping the same aspect ratio. In some cases they may not be rendered at all because the result is too small or the mobile device does not support them.
- Text and labels can be transformed as well, because they may be too long for the mobile devices. In converting labels we use tables able to identify shorter synonyms.

In particular, regarding the creation of new connections the following rules are applied:

- original connections of desktop presentations are associated to the mobile presentations that contain the interactor triggering the transition. The destination for each of these connections is the first mobile presentation obtained by splitting the original desktop destination presentation;
- composition operators that are allocated to a new mobile presentation are substituted in the original presentation by a link to the new presentation containing the first interactor associated with the composition operators.
- when a set of interactors composed through a specific operator has been split into multiple presentations because they do not fit into a single mobile presentation, then we need to introduce new connections to navigate through the new series of mobile presentations.

In the transformation process we take into account semantic aspects and the cost in terms of interaction resources of the elements considered. We define for each mobile device class identified (large, medium or small) a maximum acceptable overall cost in terms of the interaction resources utilizable in a single presentation. So in this approach, each interactor and (even each composition operator) has a different cost in terms of interaction resources. The algorithm inserts interactors into a mobile presentation until the sum of individual interactor and composition operator costs reaches the maximum global cost supported.

A SMALL EXAMPLE

In order to explain the transformation, we can consider a specific example of a desktop Web site and see how one of its pages (see Figure 3) can be transformed using our method.

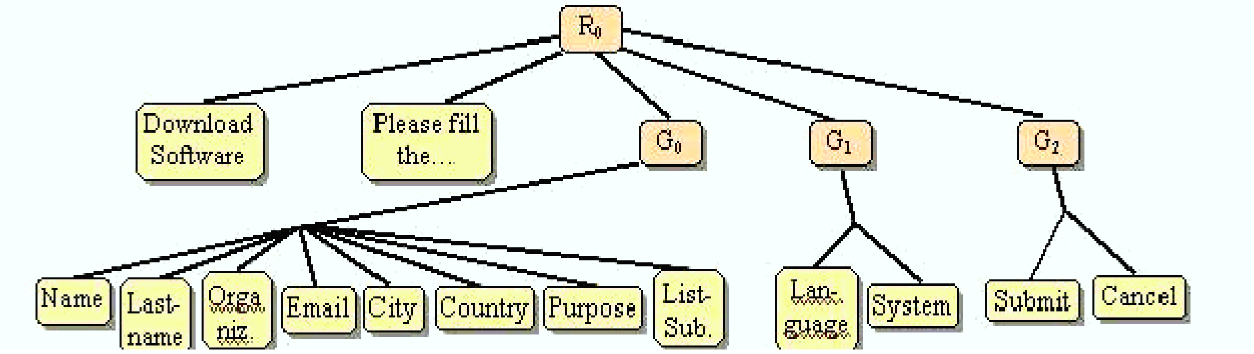
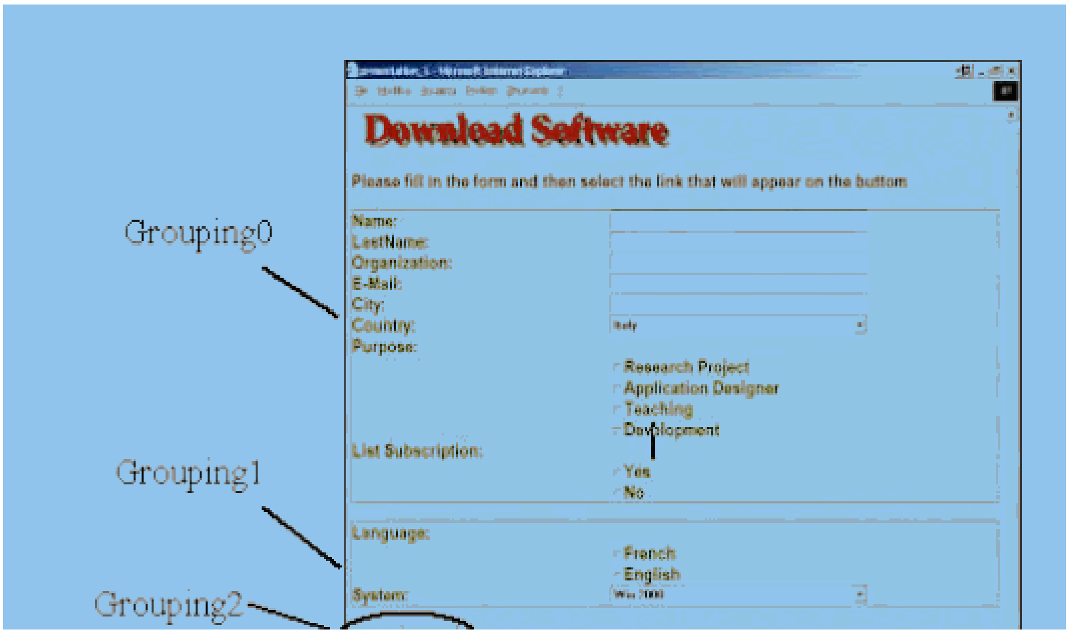


Figure 2: CUI DOM tree-structure of interface in fig 3.

The automatic transformation starts with the XML specification of the Concrete Desktop User Interface and creates the corresponding DOM tree-structure (Figure 2). The concrete user interface description contains interactors (such as text, image, text_edit, single_choice, multiple_choice, control, etc) and composition operators (grouping, ordering, hierarchy or relation) which define how to structure them. A composition operator can contain other interactors as well as other composition operators.

In the example, there is a relation operator, which involves all the elements of the page: the elementary description interactor “Download Software”, the elementary text interactor “Please fill in the form…” and the elements made up of three grouping operators. In general, the relation operator identifies an association between the last element and all the other elements involved in the operator. In this case, the last element is represented by the composition operator G2, which groups the “Submit” and “Cancel” buttons. Indeed, they are related to all the remaining content of the Web page because they allow to transmit such information to the server. There are also two grouping operators (G0 and G1) implemented by the two fieldsets in the user interface in Figure 3.

Overall, this desktop presentation contains 14 interactors, which require a large amount of interaction resources; too great to be contained in a single mobile phone presentation, even a large one (such as a smartphone). Our transformation divides the “desktop_Download” presentation of the example into four presentations for mobile devices. Considering the tree structure of the XML specification of the Concrete User Interface, the algorithm makes a depth first visit starting with the root, and generates the mobile presentations by inserting the elements in each level as long as the sum of the cost of each interactor (and related composition operators) is lower than the maximum value supported by a mobile presentation.



Download Software

Please fill in the form and then select the link that will appear on the buttom

Name:
LastName:
Organization:
E-Mail:
City:
Country: Italy
Purpose:
○ Research Project
○ Application Designer
○ Teaching
○ Development
List Subscription:
○ Yes
○ No

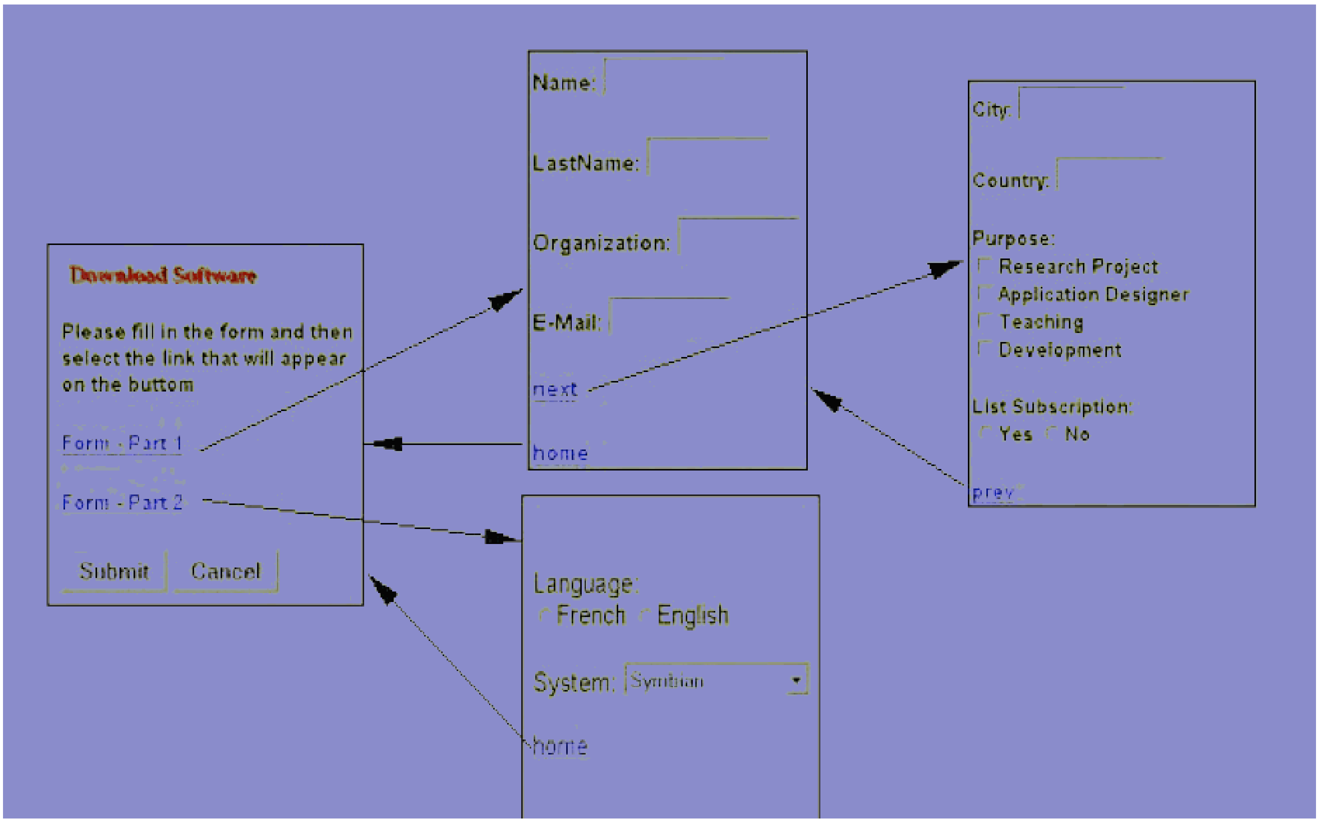
Language:
○ French
○ English
System: Win 2000

Figure 3: Example of desktop Web user interface.

Each composition operator (with its associated elements) that cannot fit in the presentation (in the example G0 and G1) is substituted by a link pointing to a mobile presentation containing their first elements. So, looking at the example, the algorithm begins to insert elements in the first mobile presentation, and when it finds a composition operator (such as G0), it starts to generate a new mobile presentation with its elements. Continuing the visit, the algorithm explores the composition operator G0, which has 8 elements, all of which cannot fit in a single new presentation. Thus, two mobile presentations are created and the algorithm distributes the elements equally between them. The depth first visit of the tree continues and reaches G1. The algorithm inserts in the main mobile presentation a corresponding link, pointing to the newly generated mobile presentation where the elements of G1 are inserted.

The relation operator captures a many-to-one association. The latter element of such a Relation must be contained in its entirety in the same presentation as the other elements of the same Relation because it is the fundamental element defining the association. Even when the latter element is another composition of elements (such as G2), it is completely inserted into the presentation (cost permitting).

The transformation creates new connections among the presentations for the mobile phone following the rules introduced in the previous section. One rule indicates that composition operators that are substituted by a link introduce new connections to presentations containing the first interactor associated with the composition operators. In the example (Figure 4), we have two new links “Form – Part 1” and “Form – Part2”, which support access to the pages associated with the first interactor of G0 and the first interactor of G1 respectively. Another rule is applied to introduce new connections to navigate through the new mobile presentations.



Download Software

Please fill in the form and then select the link that will appear on the buttom

Figure 4: Mobile pages resulting from the transformation of the example desktop page.

CONCLUSIONS AND FUTURE WORK

We have discussed issues related to consistency in cross-platform design. In addition, we have presented a solution for transforming an interface from desktop to mobile platforms in such as was as to maintain semantic consistency. Thus, grouped information in the desktop version is still grouped in the mobile version but in a manner that takes into account the more limited screen resources. The choice of the implementation techniques is still driven by the need to preserve the user interface semantics but adapting the features of the target device.

The first user tests conducted on the interfaces resulting from this approach are promising, even when this type of transformation is applied in migratory interfaces and the target user interface is dynamically created following a migration request from the user. We plan to conduct further empirical studies to compare our semantic-based transformation with other types of transformation-based approaches to cross-platform design.

REFERENCES

[1] Bandelloni, R., Berti, S., Paternò F., “Mixed-Initiative, Trans-Modal Interface Migration”, Proceedings Mobile HCI 2004, Glasgow, September 2004, Lecture Notes Computer Science 3160, pp.216-227, Sprinter Verlag.

[2] Mori, G., Paternò, F., Santoro, C., Design and Development of Multi-Device User Interfaces through Multiple Logical Descriptions, IEEE Transactions on Software Engineering, August 2004, Vol.30, N.8, pp.507-520, IEEE Press.

[3] G.Mori, F.Paternò, Automatic semantic platform-dependent redesign, Proceedings Smart Objects and Ambient Intelligence 2005, pp.177-182, Grenoble, October 2005.