Distributing User Interfaces with MARIA

Marco Manca, Fabio Paternò
CNR-ISTI, HIIS Laboratory
Via Moruzzi 1, 56124 Pisa, Italy
fabio.paterno@isti.cnr.it

ABSTRACT
The increasing interest for distributed user interfaces needs languages that better support their specification in multi-device environments. In this paper we discuss an approach to specifying this type of user interfaces, and how it is applied to an extension of the MARIA language aiming to provide flexible support for the description of distributed user interfaces.

Author Keywords
Distributed User Interfaces, Model-based User Interface Design, User Interface Description Languages.

ACM Classification Keywords
H5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

General Terms
Design

INTRODUCTION

One of the main technological trends is the steadily increasing number of devices per person. This has an impact on the user interface languages and technologies because it will be more and more common to interact with an application through multiple devices. A typical applications scenario is using a graphical or a vocal mobile device to interact with a large screen

This trend has stimulated increasing interest in Distributed User Interfaces (DUIs), which have been defined [2] as those interfaces whose different parts can be distributed in time and space on different monitors, screens, and computing platform, depending on several parameters expressing the context of use, such as the user, the computing platform, and the physical environment in which the user is carrying out her interactive task.

Model-based approaches [3] have often been considered in order to manage the increasing complexity derived from managing user interfaces in multi-device environments, since each device has specific interaction resources and implementation languages to execute such user interfaces.

An interesting approach has been proposed [6] through the use of a XML-based language called HLUID (High Level UI Description), in which the user interface has a hierarchical structure and the leaves of the tree are Abstract Interactor Object (AIOs) describing high-level interactors. During the rendering process the AIOs are mapped onto Concrete Interaction Object (CIOs) associated with the current platform. In addition, they introduce a split concept for the groupings through an attribute that, when it is set to true, allows the distribution of the user interface elements without losing its logical structure.

In our case we propose a different solution, still using a model-based approach. One difference is that we support the specification at the concrete level because at this level it is easier to generate the corresponding implementations and there is a better understanding of the actual effects that can be obtained.

In the next section we describe the design concepts that characterise the proposed solution for specifying user interfaces distributed across various devices with MARIA. Our approach is not limited to graphical user interfaces but can address also devices supporting other modalities, such as voice. Then, we also provide some application examples. Lastly, some conclusions are drawn along with indications for future work.

THE PROPOSED SOLUTION

The proposed solution has been introduced for the MARIA language [5], which provides both an abstract and a set of concrete languages. The concrete languages refer to different platforms, characterised by their interaction modalities (graphical desktop, graphical mobile with touch, vocal, graphical and vocal combination). Thus, a distributed concrete user interface should be able to indicate how the user interface elements are distributed among devices, which can belong to such various platforms.

One important point to consider is at what level of granularity the distribution can occur. We have identified four possible levels:

- Presentation, which means the set of elements that characterise the possible perceivable information at a given time (e.g. a page in a Web application);

CHI 2011, May 7–12, 2011, Vancouver, BC, Canada.

- Composition of elements, groups of elements inside a presentation that are logically related among them;
- Interactor element, single user interface elements;
- Interaction phase, we have identified three general phases that characterise interactions: prompt, input, and feedback, and it is possible to distribute the user interface even across such sub-components of one interactor. For example, entering a text through a mobile device and receiving feedback both on the mobile and a large screen device.

The next aspects that needs to be addressed is to define how distribution can occur. For this purpose we have considered the CARE properties [1], which were introduced for describing multimodal user interfaces, and have already been applied for this purpose to the MARIA language [4]. They are a simple but expressive set of properties:

- Complementarity, indicates that the user interface elements that we are considering are distributed across multiple devices;
- Assignment, indicates that the user interface elements considered are assigned to one given device;
- Redundancy, indicates that the user interface elements considered are replicated in multiple devices;
- Equivalence, indicates that the user interface elements considered can be supported by either one device or another.

The basic idea is to associate CARE properties to the various granularity levels considered. There are a few cases in which one property is not meaningful. In particular, the equivalence property mainly refers to any user input, and allows the designer to indicate that the user can choose among multiple devices for entering such input. However, the equivalence property is not meaningful for prompt or feedback phases of interaction, since such parts are controlled by the system according to its implementation and there is no possible choice for the user.

In this way we can flexibly specify distributions, even at a very detailed level. For example we can take one interactor to enter some text and specify that it should be distributed across three devices: a mobile, a vocal, and a graphical PC with a large screen in such a way that the prompt is complementary, the input is equivalent, and the feedback is redundant. This means that at the beginning the three devices will provide different prompts, which can address various aspects of the text to enter, then the user can freely choose the device to enter the requested text, and then all the three devices will present the feedback of the actual text entered by the user.

In the application of the CARE properties, we consider a user interface specification as a hierarchical structure: the user interface is composed of presentations, presentations contain single interactor elements, and compositions of such elements. In particular MARIA has various operators for composition of interactors:

The different types of interactor compositions are:

- *Grouping*: a generic group of interactor elements.
- *Relation*: when two or more interactors or compositions are related to each other.
- *Composite Description*: represents a group aimed to present contents through a mixture of *Description* and *Navigator* elements.
- *Repeater*, which is used to repeat the content according to data retrieved from a generic data source

When the CARE properties are applied to a high element in the structure then then they are automatically inherited by all its children. Thus, for example if we indicate that a presentation is redundant across two devices (one desktop and one mobile), it means that there will be two versions of such presentation, one for each device. Consequently, when from one point of the user interface there is an access to such presentation the two versions are activated at the same time. Another example is a grouping associated with a complementarity property. This means that part of the interactors in it are assigned to one device and part to another one. The corresponding device of each included interactor is indicated at the grouping level, and does not need to be specified again inside the interactor definitions also in this case.

This mechanism is an efficient way to associate flexibly devices to various user interface components without having to duplicate such device allocation in all the interface element definition.

SOME EXAMPLE APPLICATION

If we consider a composition operator, such as the grouping, then the specification of the involved interactors varies depending on the CARE property. In the case of *redundancy*, it is sufficient to list the devices in which all the grouped interactor should be supported. For example:

```
<grouping>
<output care_value="redudancy">
        <bind>
          <device id="paterno" platform="desktop"/>
          <device id="sisti" platform="vocal"/>
          <device id="iphone_lab" platform="mobile"/>
          <device id="manca" platform="multimodal"/>
        </bind>
  </output>
  <grouping>
```

The case of *complementarity* is more complex. There are various possibilities. In one case we can indicate for each interactor to what device is associated, and such interactors are not distributed in terms of their sub-components (prompt, input and feedback). Thus, in this case at the grouping level we indicate the interactor/device allocation and nothing is specified at the interactor level regarding CARE properties. This also happens in the case that there is a complementarity (thus the interactors are distributed across various devices) in which there is some redundancy (which means that in this distribution it can occur that some interactor is duplicated on multiple devices). Below you can see an example in which there is complementarity at the grouping level. Thus, one interactor (description_video) is associated with one desktop device and another interactor (description_text) is associated with three devices (one vocal, one mobile and one multimodal). Then, the corresponding concrete descriptions are provided. It is possible to note that in the case of the concrete multimodal description, CARE properties are used to indicate the allocation of the sub-components to the various modalities (in this case graphical and vocal) available on the multimodal device. Since in the example we consider an only-output object, only this aspect is assigned to one of the CARE properties (since in this case the distinction in prompt, input, and feedback does not apply given that it is only for interactive elements).

```
<grouping>
    <output care_value="complementarity">
        <bind>
            <interactor interactor_id="description_video"/>
            <device id="paterno" platform="desktop"/>
        </bind>
        <bind>
            <interactor interactor_id="description_text"/>
            <device id="sisti" platform="vocal"/>
            <device id="iphone_lab" platform="mobile"/>
            <device id="manca" platform="multimodal"/>
        </bind>
    </output>

    <description id="description_video">
        <description_desktop>
            <video src="video.flv" alt="alternative_text"/>
        </description_desktop>
    </description>

    <description id="description_text">
        <!—REDUNDANT DISTRIBUTION -->
        <description_mobile>
            <text><string>Text </string></text>
        </description_mobile>
        <description_vocal>
            <speech><content>Text </content></speech>
        </description_vocal>
        <description_multimodal output="redudancy">
            <!-- [graphical part] -->
            <text><string>Text</string></text>
            <!-- [vocal part] -->
            <speech><content>Text</content></speech>
        </description_multimodal>
    </description>
</grouping>
```

An even more complex case of distribution is the case in which the interactors in a grouping composition are distributed according to the complementarity property, and then some of such interactors are further internally distributed according to the interaction phase. In this case part of the distribution is specified at the grouping level and part at the level of the single interactors. Below an example of this case, which distributes one interactor single_choice_id to both a desktop and a multimodal device and one interactor vocal_text to one vocal device. When we move to the specification of the single_choice_id interactor, we can see that it is distributed in such a way that the input is assigned to the desktop system while the prompt and the feedback are complementary across the desktop and the multimodal device. Then, the specification of the multimodal interactor must be consistent and using the CARE properties to distribute across modalities only those aspects that have been assigned to the multimodal device in the specification of the distribution across devices. Thus, in this example only the prompt and feedback can be distributed across the modalities of the device under consideration since the input part was distributed only to the desktop device.

```
<grouping>
    <output care_value="complementarity">
        <bind>
            <interactor interactor_id="single_choice_id"/>
            <device id="paterno" platform="desktop"/>
            <devices id="manca"  platform="multimodal"/>
        </bind>
        <bind>
            <interactor interactor_id="vocal_text"/>
            <device id="sisti" platform="vocal"/>
        </bind>
    </output>

    <single_choice  id="single_choice_id">
        <input care_value="assignment">
            <devices id="paterno" platform="desktop"/>
        </input>
        <prompt care_value="complementarity">
            <devices id="paterno" platform="desktop"/>
            <devices id="manca" platform="multimodal"/>
        </prompt>
        <feedback care_value="complementarity">
            <devices id="paterno" platform="desktop"/>
            <devices id="manca"  platform="multimodal"/>
        </feedback>
        <choice_element value="ch_1"/>
        <choice_element value="ch_2"/>
        <desktop_single>
            <drop_down_list>
                <label label_position="left">
            <text_width="200px"><string>Label</string></text>
                </label>
            </drop_down_list>
        </desktop_single>
        <multimodal_single
        prompt="complementarity" feedback="redundancy">
            <drop_down_list>
                <label label_position="left"><text_width="200px">
        <string>Label</string></text></label></drop_down_list>
            <single_vocal_selection>
                <!-- PROMPT -->
        <question><content>Choose...</content></question>
```

```
                <!-- FEEDBACK -->
<vocal_feedback><content>You_choose...</content>
</vocal_feedback>
        </single_vocal_selection>
    </multimodal_single>

</single_choice>
<description id="vocal_text">
    <output care_value="assignment">
        <devices id="sisti" platform="vocal"/>
    </output>
    <description_vocal>

        <speech><content>Vocal text to speech </content>
        </speech>
    </description_vocal>
</description>
```

A similar approach can be applied also when the other composition operators of the MARIA language are considered (relation, repeater and composite description). In addition, such approach can be considered also when distributing interactors or composition of interactors at the presentation level.

EXAMPLE APPLICATION

In this section we show an example application considering a museum application in which a user plays an association game on a mobile device and at some point decides to migrate to a distributed user interface across a mobile and a large screen (see Figure 1).

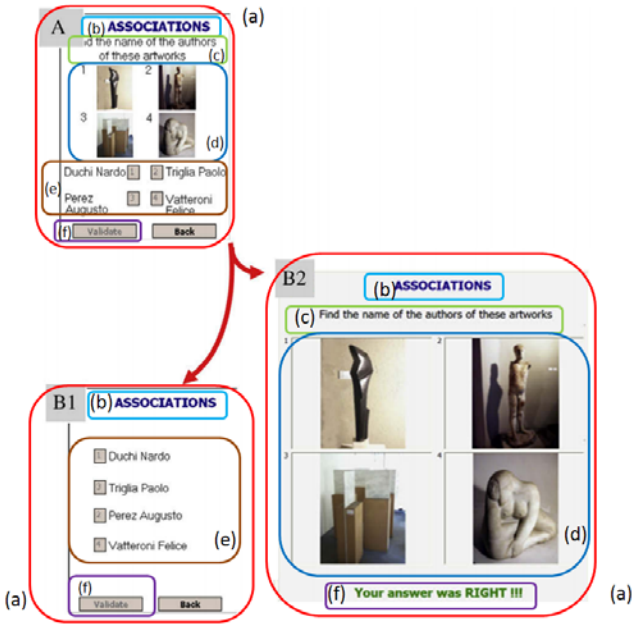

Figure 1: The example of distributed user interface.

The original grouping of elements, (a) in the figure, is distributed in a complementary way on the B1 (the mobile) and B2 (the large screen) devices. The description (text) interact (b) is distributed in a redundant manner, while the description (text) (c) is assigned only to the B2 device, along with the images (d). On the other hand, there are four text edit interactors that are assigned only to the B1 device. Lastly, there is an activator interactor (validate button) whose input and prompt are assigned to the B1 device while the feedback (the message "Your answer was RIGHT !!!") is assigned to the B2 device.

CONCLUSIONS AND FUTURE WORK

We have presented the method that we are using to develop the possibility of specifying distributed user interfaces with MARIA. Our approach allows designers to specify distribution in a flexible way, at various levels, not only at the level of interaction elements as it happens in other approaches. This is based on a logical framework, which distinguishes both the granularity levels for distribution and the properties indicating how such distribution should be applied to the considered elements. Such framework allows us to have a better understanding of the design space for distributed user interfaces, and exploit it in the design of the new features for the MARIA language.

Future work will be dedicated to supporting authoring of distributed user interfaces extending the MARIAE environment for this purpose.

ACKNOWLEDGMENTS

This work is supported by the EU ICT STREP SERENOA Project and the EU ARTEMIS SMARCOS Project.

REFERENCES

1.Coutaz J., Nigay L., Salber D.,.Blandford A, May J., Young R., 1995. Four Easy Pieces for Assessing the Usability of Multimodal Interaction: the CARE Properties. Proceedings INTERACT 1995, pp.115-120.

2. Demeure, A., Sottet J.S., Calvary G., Coutaz J., Ganneau V., and Vanderdonckt J.. The 4C Reference Model for Distributed User Interfaces. Proceedings of the Fourth International Conference on Autonomic and Autonomous Systems 2008, pp. 61-69.

3.Fonseca J.M.C. (ed.), W3C Model-Based UI XG Final Report, May 2010, available at http://www.w3.org/2005/Incubator/model-based-ui/XGR-mbui-20100504/

4.M.Manca, F.Paternò,. Supporting Multimodality in Service-oriented Model-based Development Environments, Proceedings HCSE 2010, 3rd Conference on Human-Centred Software Engineering, pp.135-148, LNCS 6409 Springer, Reykjavik, October 2010.

5. Paternò F., Santoro C., Spano L.D.: MARIA: A universal, declarative, multiple abstraction-level language for service-oriented applications in ubiquitous environments. ACM Trans. Comput.-Hum. Interact. 16(4): (2009).

6.Vandervelpen C., Conix K., Towards Model-Based Design Support for Distributed User Interfaces, NordiCHI 2004: 61-70, 2004.