A Supportive User Interface for Customization of Graphical-to-Vocal Adaptation

Fabio Paternò
CNR-ISTI, HIIS Laboratory
Via Moruzzi 1, 56124 Pisa, Italy
Fabio.Paterno@isti.cnr.it

Christian Sisti
CNR-ISTI, HIIS Laboratory
Via Moruzzi 1, 56124 Pisa, Italy
Christian.Sisti@isti.cnr.it

ABSTRACT

In this paper, we describe an approach to adapting graphical Web pages into vocal ones, and show how the approach is supported by a tool that allows the user to drive the adaptation results by customizing the adaptation parameters. The adaptation process exploits model-based user interface descriptions.

Keywords

Vocal Interfaces, Model-Based, Adaptation, Supportive User Interfaces, Accessibility.

INTRODUCTION

Vocal interfaces are important in a number of different contexts, such as for vision-impaired users or when the visual channel is busy (e.g, car driving) [7]. Design techniques in developing Vocal Interfaces has been widely studied [1] but little attention has been paid on how to adapt web pages for vocal browsing. Moreover, recognition of natural language is improving [2] and in future it will be possible to develop vocal interfaces able to recognize any user input.

We found that adaptation of graphical Web pages into vocal ones needs to be supplemented through Supportive User Interfaces (SUI), that enable the users to customize the adaptation. Indeed, a completely automatic transformation cannot provide good results in many case.

The adaptation process is based on the exploitation of MARIA [5], a recent model-based language, which allows designers to specify abstract and concrete user interface languages according to the CAMELEON Reference framework [3]. The customization tool has a Web interface allowing the user to drive the Vocal Interfaces generation.

In this workshop paper we firstly present the overall Model-Based Language Architecture, secondly we introduce the adaptation approach and lastly we show an example of application of the supportive interface for graphical-to-vocal adaptations, also showing how a parameter change can lead to different results in the final user interface.

MODEL-BASED INTERFACES in MULTI-DEVICE ENVIRONMENTS

MARIA is a model-based language, which allows designers to specify abstract and concrete user interface languages. Abstract User Interfaces (AUIs) are independent on the interaction modalities, while Concrete User Interfaces (CUIs) are dependent on the interaction resources of the target platforms but are independent of the implementation languages.

An AUI is composed by a number of *presentations*, a *data model* and a set of *external functions*. Moreover each presentation contains a number of user interface elements, called *interactors*, and a number of, so called, *interactor compositions.* Examples of interactor compositions are *grouping* and *relations* to group/relate different interactors. The interactors can be classified in terms of *editing*, *selection*, *output* and *control* and may have associated a number of *events handler*.

As already mentioned, the CUIs are dependent on the interaction resources of the target platform so, while in Desktop modality a presentation can be defined as a set of user interface elements perceivable at a given time, in the case of Vocal modality a presentation is defined as a set of dialogues between user and platform that can be identified as a logical unit (e.g. the communication necessary for a vocal form filling).

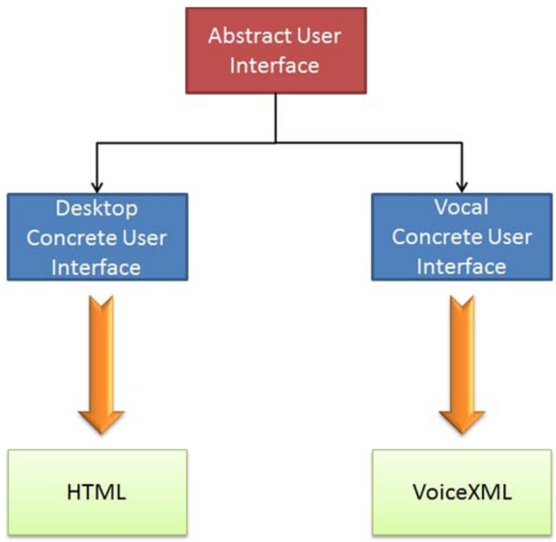

Figure 1. Some Possible Abstraction Levels

Figure 1 shows the relationship between AUI and CUIs limited to Desktop and Vocal target platform (some other target platforms available are Mobile, Multi-Touch and Multi-Modal). Figure 1 also represents some possible transformations that can be performed, such as the HTML generation from Desktop Logical

Descriptions (an instance of a Desktop CUI) and the VoiceXML generation from Vocal Logical Descriptions.

The aim of our work is to develop an adaptation process that take as input HTML pages, and generates corresponding VoiceXML (opportunely adapted for voice modality) documents. This is not a simple task and raises a large number of adaptation issues (such as the retrieving of the menu items for vocal interaction and the adaptation of images). In this context Supportive User Interfaces can provide useful support, in particular in the customization of the adaptation rules.

APPROACH

Our solution is based on an adaptation server that consists of three modules (see Figure 2):

- **Reverser:** parses the Web pages and builds up an equivalent Desktop Concrete Logical Description.
- **Adapter:** transforms the Desktop Concrete Logical Description into an adapted Vocal Concrete Logical Description.
- **Generator:** generates the VoiceXML taking in input the Vocal Concrete Logical Description.

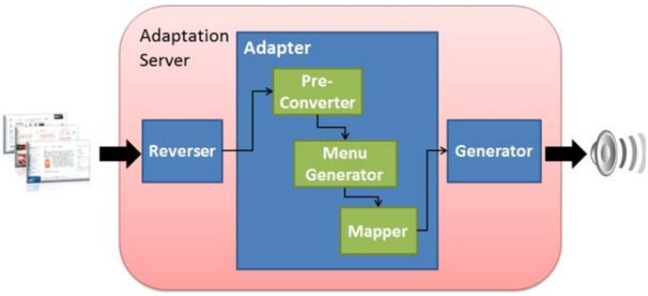

Figure 2. The Adaptation Server Architecture.

The reverser, taking into account the associated page style-sheet, transforms the HTML tag patterns into opportune Desktop CUI elements. This process enables the possibility to obtain a more semantic description. The adapter is subdivided into three sub-modules that are executed in pipeline:

1. **Pre-Converter:** removes the elements that cannot be rendered vocally (e.g., images without ALT tag) but also corrects possible inconsistences due to the reverse process (e.g., grouping containing only one interactor due to formatting purposes).
2. **Menu-Generator:** generally the vocal interfaces are navigated through lists of menus. This step aims to convert a Desktop Logical Description into a new one structured into a set of of menus/sub-menus hierarchically structured.
3. **Graphical-to-Vocal Mapper:** with this step each elements of the Desktop CUI is mapped into a (semantically equivalent) element of a new Vocal CUI.

The final implementation language is VoiceXML 2.0 [8], a standard language, supported by W3C, for the specification of Vocal Interfaces. The VoiceXML code generated by the transformation has been tested with the Voxeo Voice Browser [9] (suggested by W3C), and has passed the validation test integrated in it. More detail on the VoiceXML generation is provided in [4].

THE CUSTOMIZATION SUPPORT

The adaptation process is complex and the results depend on a number of factors, such as the structure of the Web pages in input and their conformance to the accessibility guidelines. In order to obtain better results we have designed a Supportive User Interface, which allows the user to customize the adaptation results.
The adaptation process can be driven setting a number of parameters. Such parameters can influence different states of the transformation process.
To adjust the *pre-conversion* step the following parameters are available:

- **Remove Whitespaces:** if enabled it removes the grouping that contains only whitespaces from the computation. This can happen due to graphical formatting purposes (e.g., list of “* *”).
- **Min Image Width/Height:** images under these size limits (that not contains ALT attribute) are removed.
- **Min Grouping Threshold:** in the specification provided by the reverse engineering removing grouping operators when they contain little text (below the threshold) to synthesize.

To customize the *menu generator* step it is possible to set the following parameters:

- **Max Grouping Threshold:** if the textual grouping content length is above the max threshold, then new menu items are created by splitting the original grouping.
- **Descr/Nav ratio:** to set the ratio between the description and navigator interactors in order to identify the groupings that contain a navigator bar.

Finally, to customize the *mapper* step, the parameters are:

- **Multiple Choice**: to set how the final vocal interface will perform the multiple choice. There are two solutions: *Yes/No Questions*, for every possible choice the platform will ask a Yes/No confirmation to the user; *Grammar Based*: the user can select more than one possible choice with one single sentence (listing the choices in sequence).

- **End Form Sound:** to decide if each vocal dialogue should terminate with a short sound.

Figure 3 and 4 show our Supportive User Interface that allows such parameterization. The left panel (shown in Figure 3) contains some modifiable parameters and their default's values.

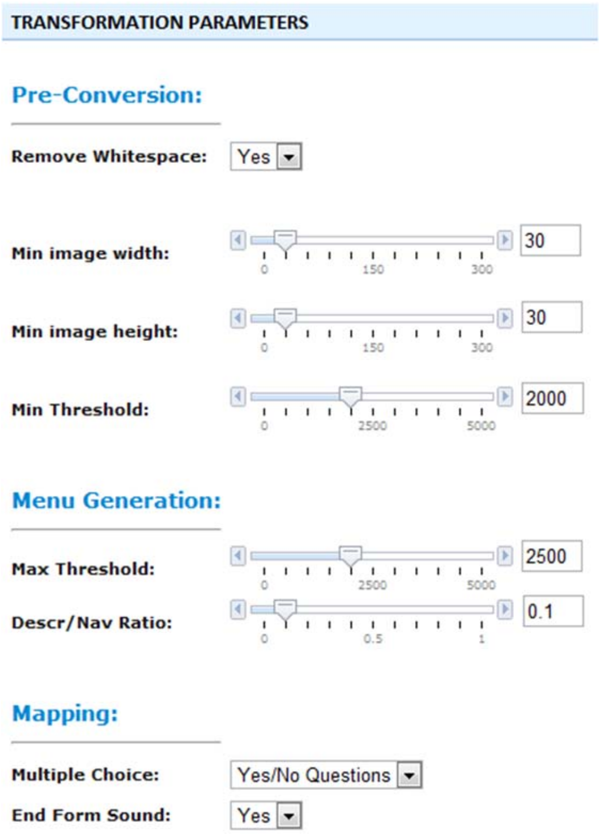

Figure 3. Customization of the adapter.

The right panel (see figure below) shows the structure and the menu items of the generated vocal page. In this way the designer can decide whether to download the final vocal interface (as a zip file containing the VoiceXML documents) or change the transformation parameters in order to obtain a different structure.

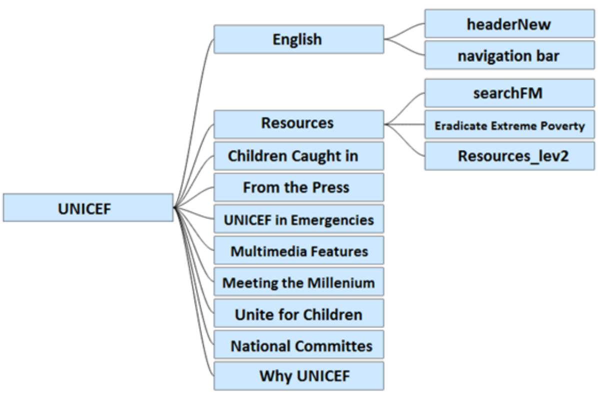

Figure 4. Application right panel: vocal menu structure.

EXAMPLE CONFIGURATION PARAMETER CHANGE

In this section we show an example of configuration parameter change, which affects the structure of the resulting user interface.

In particular, we consider Max_Threshold parameter, which defines the threshold in terms of text length to render vocally. If the length exceeds this limit the adaptation system splits the presentation content. If we set **max_threshold = 2500** then we obtain the structure in Figure 4.

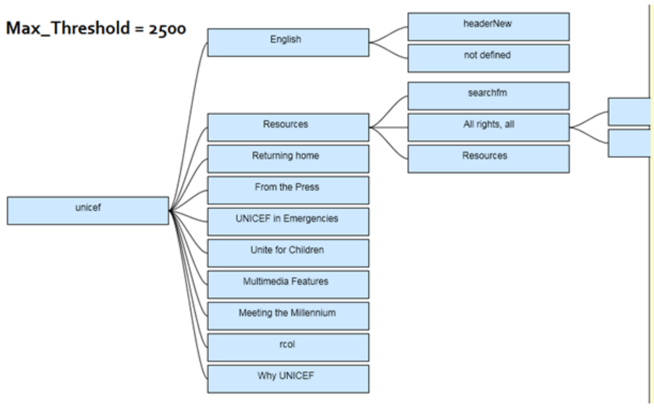

Figure 4. Initial parameter set.

Thus, the **Returning home** part (see Figure 5) will be rendered a single piece of information.

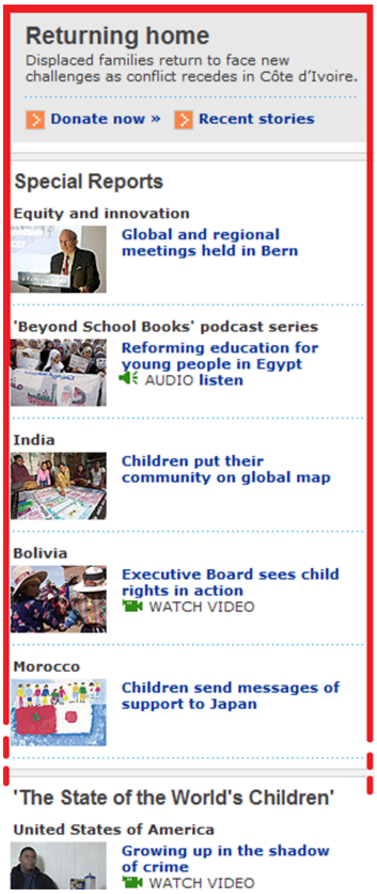

Figure 5. The considered content part.

If we change the parameter to **max_threshold = 700** we obtain the structure in Figure 6.

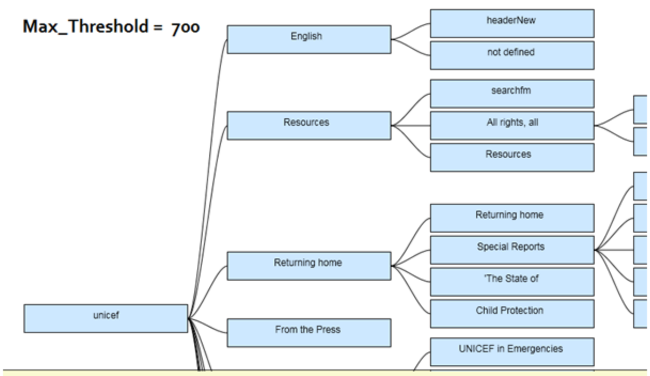

Figure 6. The resulting modified structure

We can note that the resulting structure has more sub-levels: the section **Returning home** is subdivided in multiple parts, highlighted by dashed lines in Figure 7, which can be further subdivided.

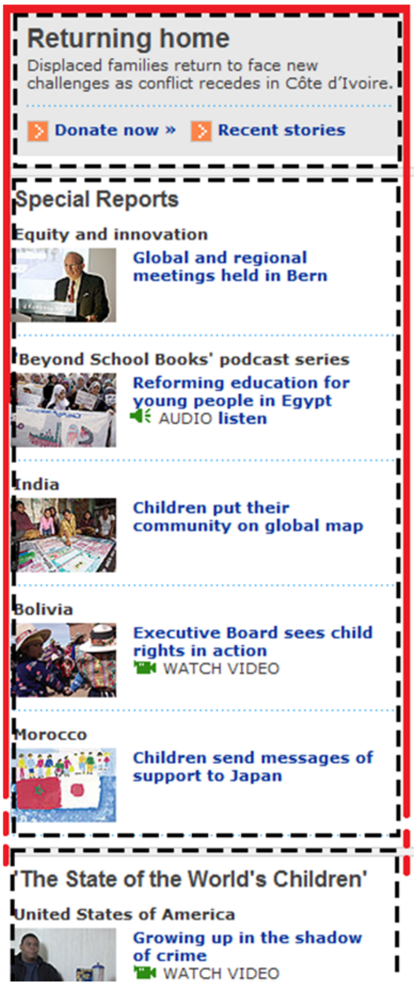

Figure 7. How the content is further divided.

CONCLUSION

A Model-Based approach to supporting Graphical-to-Vocal Adaptation is introduced. A Supportive User Interface is then proposed (as Web Application) in order to help the user to manage the overall adaptation process.

We consider this tool as useful support to provide users with full control on the final results. Given the complexity of the existing Web content, we plan to add new features to both the adaptation rules and the customization interface, in order to have further flexible control on the adaptation results.

ACKNOWLEDGMENTS

We gratefully acknowledge support from the Artemis EU SMARCOS and the ICT EU SERENOA projects.

REFERENCES

1. A., Edwards and I., Pitt.: Design of Speech-Based devices. Springer (2007).
2. A., Franz. and B., Milch.:Searching the web by Voice. In proceeding of the 19th international conference on Computational Linguistic - Volume 2, pp. 1-5, Stroudsburg, PA, USA. (2002).
3. Calvary, G., Coutaz, J., Bouillon, L., Florins, M., Limbourg, O., Marucci, L., Paternò, F.: The CAMELEON reference framework. CAMELEON project, Deliverable 1.1. (2002).
4. F., Paternò and C., Sisti.: Deriving Vocal Interfaces in Multi-device Authoring Environments. In Proceedings of the 10th International Conference on Web Engineering, pp. 204-217 (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. UNICEF. http://www.unicef.org/.
7. Voice Browser Activity. http://www.w3.org/Voice/.
8. Voice extensible markup language (VoiceXML) version 2.0. http://www.w3.org/TR/2009/REC-voicexml20-20090303/7.
9. Voxeo Voice Browser. http://www.voxeo.com/.