Cicero Designer: An Environment for End-User Development of Multi-Device Museum Guides

Giuseppe Ghiani, Fabio Paternò, and Lucio Davide Spano

ISTI-CNR, HIIS Lab, Via Moruzzi 1,
56124 Pisa, Italy
{Giuseppe.Ghiani,Fabio.Paterno,Lucio.Davide.Spano}@isti.cnr.it

Abstract. This paper describes the design and implementation of a tool to allow people without programming experience to customize the functionality and user interface of a multi-device museum guide. It consists of a direct-manipulation visual environment that supports editing of the main features of a museum guide and the creation of the associated interactive games. The tool then generates application versions for access through both mobile and large screen stationary devices. We also report on a first empirical evaluation carried out with museum curators.

Keywords: End User Development, Multi-Device User Interfaces, Mobile Museum Guides.

1 Introduction

End-User Development (EUD) [8] has focused mainly on desktop applications. However, mobile technology has penetrated many application domains and mobile devices are more and more powerful in terms of processing and interaction resources. There is an increasing number of applications that aim to exploit such technological offerings. Non-professional developers already have difficulties in developing applications for desktop systems, and targeting multi-device environments is too complex, unless they are adequately supported [4]. The prototype described in the paper is an example of a domain specific EUD environment. The identification of key semantic building blocks and target scenarios guided the creation of an intuitive metaphorical tool to configure context-sensitive museum guides, including educational games and multi-device deployment.

In particular, we consider the museum application domain, in which software applications are increasingly used to assist visitors in accessing the relevant information. In addition, museums are dynamic entities and often change the items on exhibit or their locations. Thus, it is important to allow their curators, who presumably have no programming experience, to be able to (re)configure the mobile guide, its content and interactive behaviour.

Our work aims to allow museum curators to easily create and modify guides accessible through both mobile and large screen stationary devices, providing a rich set of

V. Pipek et al. (Eds.): IS-EUD 2009, LNCS 5435, pp. 265–274, 2009.

interactions with the museum information, including some interactive games, which can be useful to improve and assess the learning experience.

2 Related Work

A visual strategy for developing context-aware applications was proposed in [7]. Such a system, called iCAP, allows end-users to design application prototypes by defining elements (objects, activities) and rules (associations between actions and situations). The rules are graphically edited through basic operations like dragging the defined elements onto rule sheets. Another framework to support people without programming experience is eBlocks [6]: it facilitates the creation of customized sensor-based systems and the configuration of condition tables.

Differently from iCAP and eBlocks, which are not specifically dedicated to end user development for mobile environments, our investigation is focused on solutions for facilitating the management of content and the associated interactive functionality also on mobile devices (namely, PDAs and smartphones).

Akesson et al. [1] present a user-oriented framework to ease the reconfiguration of ubiquitous domestic environments. The support, running on a tablet PC, adopts a different paradigm, based on jigsaws.

Carmien and Fisher [5] describe a framework for customizing mobile applications to help people with cognitive disabilities. A graphic editor, intended to be used by the caretakers, facilitates the management of the task-support scripts for helping the disabled. The evaluation of the editing environment, called MAPS-DE, revealed that the caretakers appreciate the possibility of customizing the prompting system for the needs of individuals with specific disabilities. Like MAPS-DE, our environment also allows the customization of mobile solutions, but it has educational purposes rather than disabled support and it also allows the generation of application versions for stationary systems with large screens.

The use of educational games on mobile phones for enhancing scholars' visit of archaeological sites is treated in [2] which, however, does not deal with the development and modification of application content and behaviour.

Bellotti et al. [3] propose a framework for developing edutainment applications, such as mobile tourist guides. The paper also deals with the issues related to the interaction between the user and the mobile device when rich multimedia content is presented, but it does not provide solutions for end-user development.

Some ideas regarding general environments for end-user development of multi-device interactive applications are in [4] but such ambitious goal has not found a definitive solution. In this work we focus on a specific application domain (museums) and present a solution that can be applied to other domains as well, and which can be extended to support adaptation to a broader set of devices.

3 The End-User Development Environment

In order to facilitate content creation for the guide of a new museum and/or changes to an existing one and the associated interactive behaviour, we have developed a specific visual environment for the desktop PC. The guide editor tool (as well as the resulting mobile and stationary versions of the guide) has been written in the .NET C# language. This tool accesses an XML-based description of the museum, which defines rooms, their layout, and artwork positions as well as additional information. Starting with such data, which includes the photos and descriptions of the artworks, the editor allows users to:

- Create museum rooms or sections by simply drawing polygons on the overall museum map;
- Create links for navigation among rooms using icons (e.g.: arrows or stairs) or text boxes;
- Add, remove or change artwork icons and select the associated photo, information, video and text files, used by the TTS (Text-to-Speech) engine to create the vocal comments on the fly. Each artwork can be associated with a tag (RFID, in our case) for automatic user localization purposes at run time (see Figure 1). The tag ID inserted in the editor is basically a string and the editor functionality is independent of the localization technology actually used by the guide application: the matching between the detected tag(s) and the associated artwork(s) is solved at run time when a new tag event occurs. This type of event is triggered by the localization module that interfaces with the hardware;
- Create help sections;
- Insert interactive regions on the overall museum map for quick room selection (allowing the user to manually change room by clicking them);
- Create instances of educational games;
- Insert, by drag-and-drop, game instances, which are associated with specific artworks.

3.1 The Museum Maps

Figure 1 shows the interface for editing the museum virtual environment. The rooms and the associated items are listed on the left side in a tree structure (elements can be expanded for editing). The same strategy is used in the right panel for listing the available resources (e.g.: photos). The central part is dedicated to the room currently being edited. New elements can be added to the room by just selecting the corresponding icons from the toolbox and locating them in the museum map through drag-and-drop.

After saving the configuration, the tool generates a collection of XML files, which define the corresponding database and can be simply deployed on the devices (mobile and stationary). The two database versions differ mostly in the detail level of the multimedia resources. The stationary device package contains pictures and videos with higher quality than the mobile one. In this way it is possible to exploit the better resolution of the large screen (e.g.: for items preview) and to save storage space on the mobile device. On the guide application at run-time the information available is presented differently depending on the type of device (thus, for example, long descriptions are presented only on request on the mobile, while they are immediately rendered on the large screen).

Currently, the rules determining how the user interface will appear in the two different platforms are pre-defined. In future work they will be generated from logical descriptions taking into account the capabilities of the target devices.

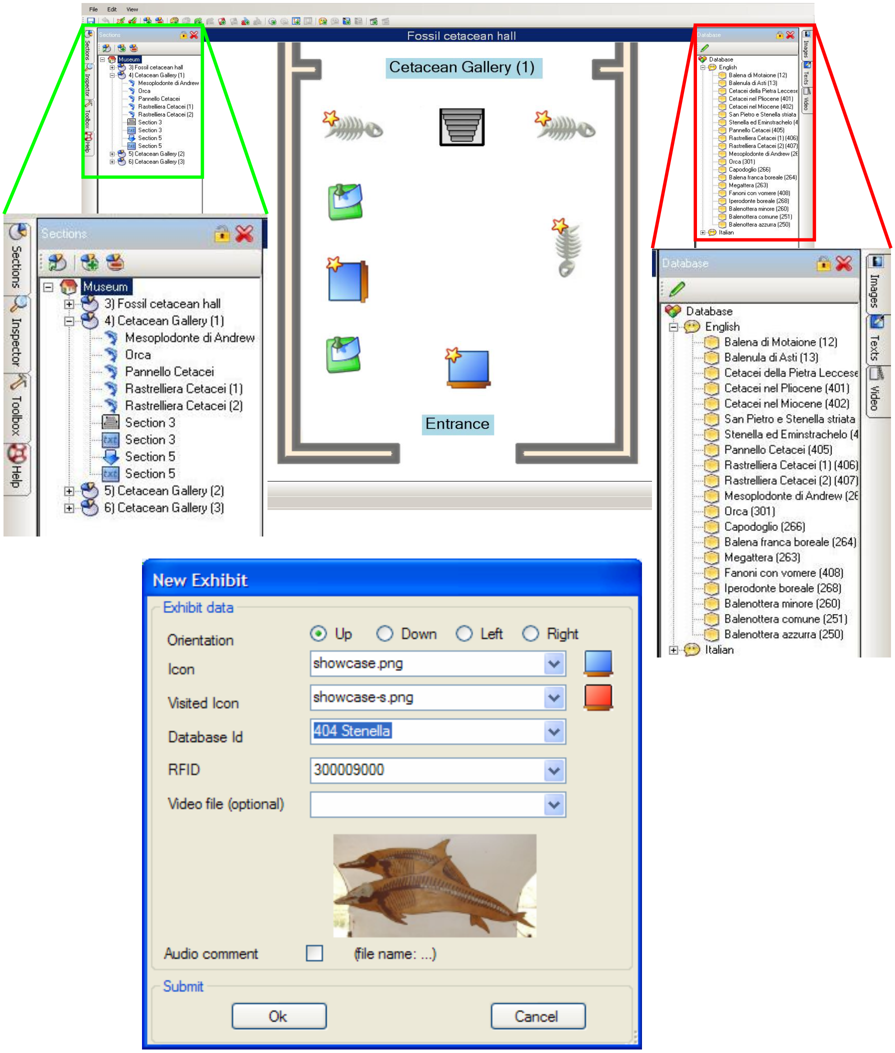

Fig. 1. The main window of the museum maps editor (top) and the form for setting the exhibit parameters (bottom)

3.2 The Games

The environment supports six types of individual games. Figure 2 shows the user interface for each:

- *Associations* requires the player to link images to information items, e.g. the picture of an artwork with its name.
- *Details* shows an enlargement of a small portion of an image. The player has to guess which of the items the detail belongs to.

- *Chronology* requires the user to order the images of the artworks shown according to their creation date.
- In the *find the word* game the user is requested to guess a “hidden word” related to an exhibit attribute: the number of characters composing the word is provided as a facility.
- In the *memory* game, the user has to observe an image for a while. After the image has disappeared s/he has to answer a question related to the image.
- The *quiz* is a single-answer multiple-choice question.

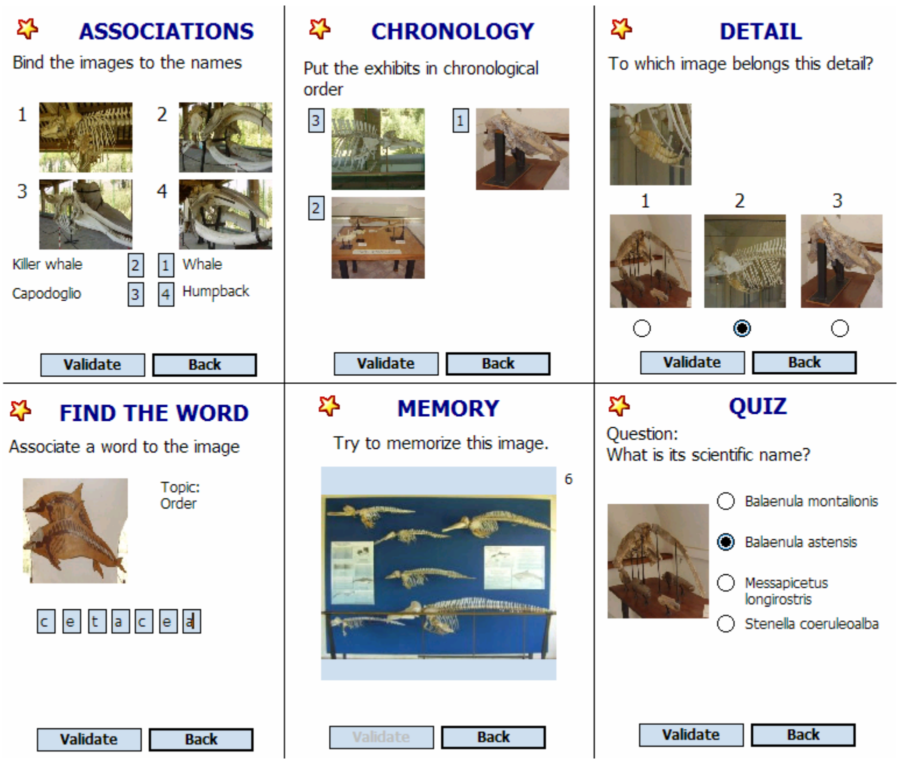

Fig. 2. The six individual games displayed on the PDA

We chose these types of games with the aim to enhance the museum visitor’s experience without interfering with the visit: users need not spend time in understanding the game rules, but should exploit the museum information in order to find the solution. For this reason the games are simple, and the difficulty depends mostly on the content.

The interface for editing a game has been designed to look like a preview of the corresponding game: the user enters the questions, the images and the captions, and provides the solution for the game.

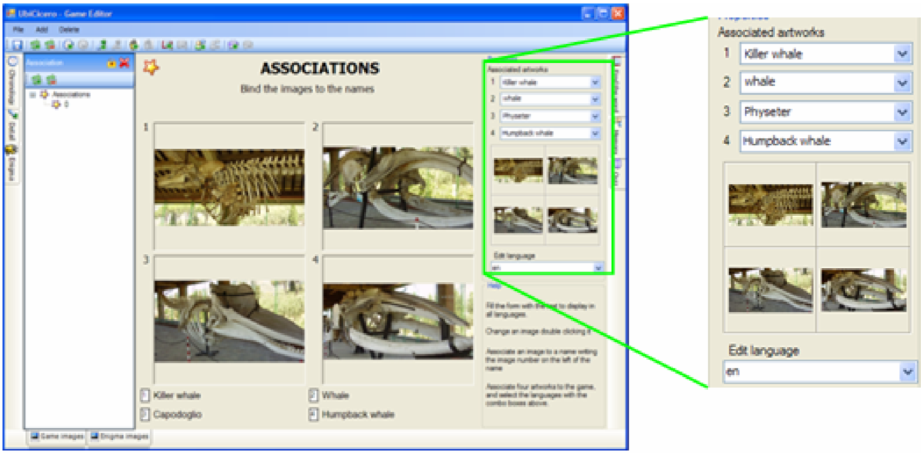

Fig. 3. The window for creating an “associations” game

To create a game, the user selects the game type (associations, chronology etc.), then associates the artwork (or the artworks if the game involves more than one, i.e. associations and chronology), and finally provides the proper content. After creating the game, a star will appear over the corresponding artwork icon. It is possible to associate a set of games to an artwork.

Figure 3 shows an example of creating an association game: the environment prompts the user (i.e., the museum curator) to provide the images, the corresponding names and the relations among them. Although every game is bound to one or more artworks, the game is not a field of the artwork data structure, that is to say it is not contained in the artwork definition. Thus, the game generation is a more dynamic process than a simple content addition. Indeed, each new game consists of a set of resources (texts, pictures) contained on, or referred by, a piece of XML code compliant to the specifications of the template. The association between artwork and game is then solved by the editor preview form, as well as by the guide application, to enable the graphic engine to draw the game icon. At run time, museum visitors may try the possible associations and receive the corresponding feedback whether the answers are correct or not.

This high configurability of the environment has made it possible to create a guide for the Natural History Museum of Calci involving one collaborator of our laboratory for about one week. The availability of an EUD tool for museums is judged important, especially after the guide application deployment, since the layout of a museum can often change. The museum curators can thus directly update the data and functionality of the guide, even without knowing the underlying implementation language.

4 The Resulting Application

The resulting application can be used through either a PDA or a desktop system, including desktops with large screens. While the two application versions have a similar logic, their user interfaces vary in order to better adapt the different screen sizes (3.5 and 42 inches). In addition, users can dynamically transfer part of the user interface from the mobile to a large screen when one is nearby in order to opportunistically exploit its better resolution or share information and comments on it with other visitors. Figure 4 shows how, after migration to the stationary device, the user can access the content that was available on the PDA through richer presentations. On the PDA, the map items are icons (A1), while on the large screen images are used (A2). The artwork/exhibit preview on the desktop (B2) has a picture with a better resolution than in the PDA (B1), and the text is shown in its entirety (while on the PDA the description is provided vocally) and the position of the visited section with respect to the museum map is presented as well.

Thus, the user interfaces differ depending on the mode in which the application is accessed: mobile only, desktop only, distributed across the two platforms.

In the case of mobile access:

- The item has a label, a low resolution photo (about 150x200 px), chronology and a summary of main information (e.g., authors for artworks or scientific names for animals). The description is automatically read by text to speech software;
- The games are represented by title and description (or question, if any), item photo, UI for answering through PDA-designed interaction (e.g., four clicks in sequence for chronology order);
- Museum visit consists of presenting section/room maps with low resolution icons of the items.

In the case of stationary access through large screens:

- Items are presented by name, high resolution photo (about 800x600 px), detailed description, context information (life of the author and historical descriptions for artworks, species information for animals etc.), related items, position in the museum and path from current position.
- Games are represented as in the PDA, with high resolution photos of the items and standard desktop interactions.
- Museum visit is supported through high resolution section or room maps with the possibility to change the display of the items using icons, previews or both, and whole museum map with current position, as well as collaborative game status

In the case of distributed user interface:

- The PDA shows game title, description (or question) and the UI for answering through a PDA-designed interaction technique;
- the desktop shows same title, description and high resolution photo of the items

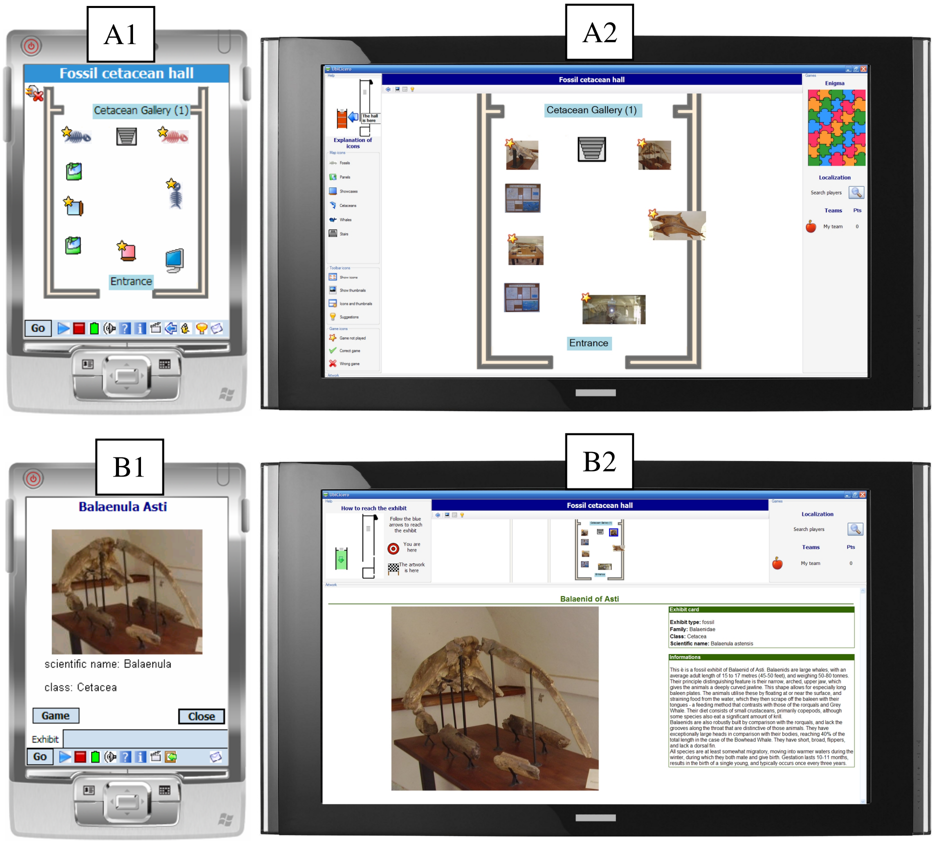

Fig. 4. Device dependent representations: virtual section on PDA (A1) and large screen (A2); artwork preview on PDA (B1) and large screen (B2)

5 User Test

We performed a first user test of our end-user development platform for museum environments. In order to get feedback from the target users, we involved the curators of the Museum of Natural History in Calci (Italy). Two of them participated in the test session. Both had several years of experience on personal computers and applications, however they had never used any environment for managing museum content or, more generally, visual builders.

Before starting the test, the participants viewed a short demonstration on the capabilities of the tool and received explanations on the main components. Then, they were provided with the list of tasks to perform:

- Set up one room (reflecting the real layout)
  - Extract the room map from the global museum map
  - Create, configure the exhibits and put them on the map
  - Insert the link items for section changes

- Add a new item to an existing room
- Generate two educational games and bind them to existing exhibits.

Both users were able to accomplish the assigned tasks. After performing the test, the users were requested to answer a questionnaire providing ratings (1 to 5) as well as subjective opinions and suggestions for improving the tested solution. Users rated the tool's features 4 and 5, and provided positive feedback. The museum curators appreciated the possibility of quickly editing the museum guide descriptions, not only for setting up new sections but also for managing the rooms whose layouts often change. This was considered a key feature for a museum guide: the curators often change the presentation of exhibits or artworks (for special exhibitions, artists celebrations etc.), and a software without this possibility would soon be considered obsolete.

One user reported that the ease of editing would enable easily creating ad-hoc games for teachers in order to evaluate their knowledge before actually accompanying students to the museum. The other user suggested enhancing the game editor interface with previewing capabilities, in order to let the user immediately see the actual presentation provided to the visitors on the PDA.

6 Conclusions and Future Work

End-User Development has mainly focused on desktop applications, but people use more and more mobile devices as well in order to access interactive applications. In this work we present an environment to support end-user development of museum applications that can be accessed through both mobile and desktop devices. This work shows a systematic design pattern: domain analysis to define target functionality and building blocks for component-based design time, use of direct manipulation design principles, templates (in this case game patterns, for example). These principles can be transferred to other domains as well.

Future work will be dedicated to the possibility of extending the approach in such a way to generate application versions accessible also through different modalities (such as voice) exploiting the use of XML-based declarative descriptions of user interfaces and a set of adaptation rules customizable by users. We also plan to conduct further empirical validation of the approach proposed.

References

1. Akesson, K.P., Crabtree, A., Hansson, P., Hemmings, T., Humble, J., Koleva, B., Rodden, T.: "Playing with the Bits" User-Configuration of Ubiquitous Domestic Environments. In: Dey, A.K., Schmidt, A., McCarthy, J.F. (eds.) UbiComp 2003. LNCS, vol. 2864, pp. 256–263. Springer, Heidelberg (2003)
2. Ardito, C., Buono, P., Costabile, M.F., Lanzilotti, R., Pederson, T.: Mobile games to foster the learning of history at archaeological sites. In: VL/HCC 2007, pp. 81–86 (2007)
3. Bellotti, F., Berta, R., De Gloria, A., Margarone, M.: MADE: Developing Edutainment Applications on Mobile Computers. Computers & Graphics 27, 617–634 (2003)
4. Berti, S., Paternò, F., Santoro, C.: Natural Development of Nomadic Interfaces Based on Conceptual Descriptions. In: End User Development, pp. 143–160. Springer, Heidelberg (2006)

5. Carmien, S.P., Fischer, G.: Design, Adoption, and Assessment of a Socio-Technical Environment Supporting Independence for Persons with Cognitive Disabilities. In: Proc. CHI 2008, pp. 597–606 (2008)
6. Cotterell, S., Vahid, F.: A Logic Block Enabling Logic Configuration by Non-Experts in Sensor Networks. In: Proc. CHI 2005, pp. 1925–1928 (2005)
7. Dey, K.A., Sohn, T., Streng, S., Kodama, J.: iCAP: Interactive Prototyping of Context-Aware Applications. In: Fishkin, K.P., Schiele, B., Nixon, P., Quigley, A. (eds.) PERVASIVE 2006. LNCS, vol. 3968, pp. 254–271. Springer, Heidelberg (2006)
8. Lieberman, H., Paternò, F., Wulf, W.: End-User Development. Springer, Heidelberg (2006)