

A Proposed Architecture for the Generation of Adaptive Interfaces in Mobile Devices

Giani Carla Ito¹, Maurício Ferreira¹, Nilson Sant'Anna¹

¹Laboratory for Computing and Applied Mathematics - LAC

Brazilian National Institute for Space Research - INPE

C. Postal 515 – 12245-970 – São José dos Campos - SP

BRASIL

E-mail: {giito, nilson} @lac.inpe.br, mauricio@ccs.inpe.br

Abstract. *The proposal of this article is to present an architecture for Generation of Adaptive Interfaces (GAI) which aims at creating user interfaces adjustable to the characteristics of the mobile devices in run time, considering the classes of devices, the platform types, aspects of user contexts, among other factors. It was used one way of identifying mobile devices proposed by World Wide Web Consortium (W3C), named Composite Capability Preference Profile (CC/PP).*

Keywords: adaptive interfaces; CC/PP; mobile computing

1. INTRODUCTION

The increase of mobile devices diversity and its use in total dynamic environments for some types of users and the adaptation of an interface to this new context became a necessity. The development of interfaces for mobile devices involves some challenges, as heterogeneous environments, physical limitations of the device among others. The adaptive interfaces present themselves promising in the attempt to surpass the current problems of complexity in the interaction of man and computer. To improve this interaction, it is necessary interfaces that are capable of adjusting themselves to the user necessities, where the system analyzes this user's actions and profiles adapting itself automatically.

The reusability of code is a basic aspect on this context. The reuse of interfaces is complex to the appearing of new models of devices with several characteristics. To make more flexible this process it is necessary to create and to use development forms that are less dependent on the properties of a specific device.

So that a site can be visualized either in personal computers or in mobile devices where the developer can choose one of the following approaches:

- To develop a site for each type of device;
- To develop only once the code of a site and adapt it through the generator to be shown in other devices.

On the first approach, the effort demanded on the developer part is great, therefore it must model a site for each device that can effect the request. On the second approach, the site is only modeled once and the generator of interfaces based on the information of the device that effects the solicitation, generates the code, so the layout is adapted in agreement with the device screen. In contrast to the previous approach, on this one occurs the reusability of code, once it will not have the necessity that several sites would have to be remade.

This work aims to present the proposal of architecture for the Generation of Adaptive Interfaces (GAI) which has as objective the creation of interfaces that adapt themselves to the characteristics of mobile devices, arising from a generic model that answers the requests of diverse users for only one application. It has as goal dynamically return the elements from the interface in accordance with the distinct characteristics of each device.

It will be used a form of recognition for mobile devices proposed by the Consortium World Wide Web (W3C), named Composite Capability Preference Profile (CC/PP) which has the purpose of providing a structuralized and universal mechanism to describe and transmit information about the characteristics of a mobile device for a server. It will be presented tests with the repository of stored data locally.

2. ACQUIRING DEVICES CAPABILITIES

The Composite Capabilities/Preferences Profile (CC/PP) is a specification of the W3C to express characteristics of the devices and the preferences of users. The profile CC/PP is a description of the device potentialities and the user preferences, Hanumansetty (2004).

It has as objective to provide a structuralized and universal mechanism to describe and transmit information about the capacities of a Web client for a server so that the content is directed to these characteristics. The profiles can be stored in a local or remote Web server, Butler (2002). Each form of storage presents advantages

and disadvantages, however the use of both in one same application can become the recognition of more efficient, productive and faster mobile devices.

The CC/PP is based on Resource Description Framework (RDF), a language used for the W3C for modeling of metadata and document description eXtensible Markup Language (XML), allowing a bigger flexibility for the creation of new vocabularies, Sila and Swick (1999).

The profile CC/PP is constructed in a hierarchy of two levels, the first, is named components of a profile and the second as attributes of a profile. A component is composed of at least one attribute, which can be hardware, software or browser. As a hardware example there is the size of the screen, processor, memory, etc. About software, the version of the operational system and browser name, version, fabricant, etc. The use of the architecture where the profile is stored locally has as main advantage to the speed of reply on the request of profiles CC/PP.

However the database of profiles becomes obsolete due to the quickly appearing of new devices. So that the recognition of devices becomes efficient and it is necessary that for each new device its profile would be added in the local repository. In the architecture where the profiles are in remote servers, there is the benefit of the constant update of the fabricants, but the access is slower.

3. ARCHITECTURE FOR GENERATION OF ADAPTIVE INTERFACES (GAI)

The environment of the mobile computation creates the necessity of a bigger flexibility of software imposing new challenges for the modeling and development. Currently diverse studies are being carried out in search for more dynamic interfaces, that help the user realize several activities in a more pleasant and efficient way.

The architecture GAI was constructed based on Model View Control (MVC), that establishes a separation from the structure of the interface in three distinct parts: Model, Vision and Control. The vision manages the graphical and textual output visible for the user. The Control interprets the input of mouse and keyboard, commanding the Vision and the Model to alter themselves of appropriate form. The Model manages answering the requests about its state, to the instructions the behavior changes and also the domain data of the application.

The studied works, MUSA by Menkhaus (2002), and Dygimes by Coninx and Luyten (2003), had used the concept of separation for layers, having made a clear distinction between the model of implementation of each level: interface of the user and logical model. Sendin and Lores (2004) affirm that the MUSA and Dygimes are limited, therefore these consider aspects as logical model, users and models of platforms, leaving without modeling the contextual aspects, except in some specific cases.

For the authors, architectures for the anticipation of contextual changes, are the ones of the main gaps in literature. For this fact, the authors developed an architecture that provides a dynamic adaptation, using a reflexive architecture. In this model, the separation among concepts is used, where the developer aims only on the functionality of the application, without worrying about the interface. This way, it is emphasized the reusability of code, where an application can be used with multiple interfaces.

The proposal of architecture GAI is the generation of interfaces that adapt themselves to the mobile devices characteristics, using the description of a generic interface. It has as main purpose to attend the requests of diverse users for only one application and dynamically return elements from the interface according to the distinct characteristics of each client on real time, considering the type of mobile device and the context of the user. Another prominent factor of the architecture is to allow an implementation that is directed to a multi-platform environment where only one application can generate many interfaces dynamically. The main advantages of the architecture GAI use can be:

- The adaptation process of the interface to the type of device that will happen in real time;
- Easiness for the development process of interfaces, therefore the developer will create just one interface that will be presented in multiple devices;
- The interface designer and the application developer can work separately in the project of a system;
- It will have the device recognition thought automatic form;
- The architecture will provide easiness of code maintenance, since the alterations (in case that new models appear) could be made updating only the metadata ones;
- The architecture will be used through a graphical editor who will supply an environment of development with visualization in the emulator of the requested device, as well as a metadata manager;

The interface can be generated for diverse platforms and languages, being based on the metadata's.

The figure 1 illustrates composed architecture GAI for two levels: client and server. Users can request services provided with many types of PDAs amongst them Palms, Pocket PCs or smart phones. In the server are located, metadata's proxy of interface and interface as follow:

- **Metadata:** They will be stored in a database containing the user profile, the interface description and the components of the interface description;
- **Proxy of Interface:** It contains the control that retrieves the metadata, the device profile and the

generator of interface that is the responsible one for the adapted interface generation;

- **Interface:** It contains the interfaces of the application. The first one will be adapted and the others will be adapted. In case that the profile in the mobile devices is not found all the interfaces will have the system standard.

To illustrate the use of architecture GAI, it is assumed that the user requests some type of service to the server. This service will be sent through the layer of communication Hyper Text Transfer Protocol (HTTP). The mobile client through browser directs for the control layer the heading HTTP. The control retrieves the profile the device in the repository CC/PP and user profile if the same exists. In case that it does not exist a standard profile is considered.

The control then analyze the specification the interface and the language definition supported by the device, e. g., Hypertext Markup Language (HTML), Extensible Hypertext Markup Language (XHTML) or Wireless Markup Language (WML) and sends these information to the "Generating Interface" that will implement an adaptation algorithm carrying through the alterations, modifications and appropriate render and consequently will generate in the interface for the mobile device returning it for the client.

If the device profile requested by the service is not found, the system will return an standard interface. This process can be visualized on figure 1. Through this procedure, it will not be necessary the code development every time that changes will happen in the domain, once the alterations can be gotten in the database stored models through the metamodel that will be implemented.

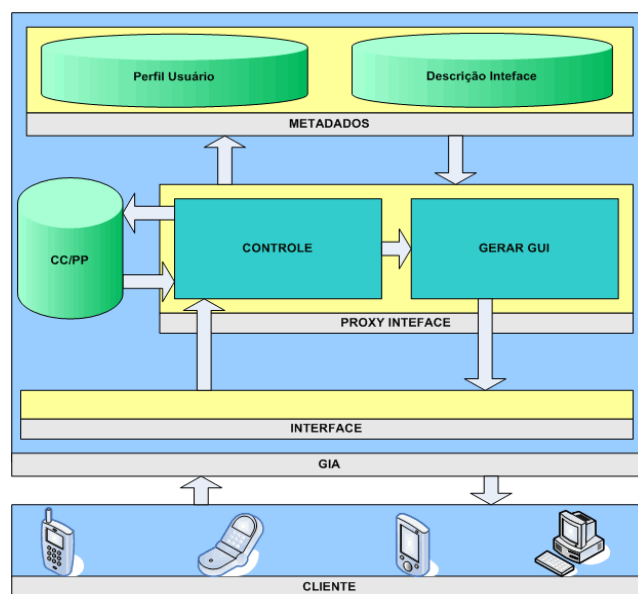


Figure 1. Architecture for Generation of Adaptive Interfaces (GAI)

4. GAI LAYOUT MANAGEMENT

The adaptation process does not guarantee that an interface keeps the usability principles. It must be carried out in a way that minimizes the impact up on the user. On the other hand it is necessary to guarantee that any modification on the interface keeps its usability.

In case it is necessary to reduce the shown content, the process of a data choice should be made carefully, not occulting or removing important parts that will be necessary for the understanding, execution and finishing of some tasks. Next some requirements related to the layout management are described and will be applied in architecture GAI for the interface adaptation guarantee.

4.1. Screen Mapping

The interface will be divided in regions to facilitate the adaptation process. Each region could also be composed

for sub-regions and for visual components as texts, buttons, links, combos, etc, that could be located in them. The figure 2 illustrates a site interface composed by some components, while the figure 3 shows its division in regions. The regions could be divided or not, Vermeulen (2005). All the components came from the same group, on hierarchical form, keeping a logical relation between them in accordance with the functionality determined by the developer, as shows figure 4.

4.2. Layout Management

It is the process to determine the size and position of the visual objects that are part of the interface. Two restrictions will have to be applied:

- **Re-dimension:** used to modify the size of images and components. The minimum width and height can be kept and also the ratio of the object when the same will be diminished. It must be verified if this is legible or not.
- **Displacement:** used to modify the figures and components localization. When it is desired to prevent the use of horizontal scrolling bar. Also it is used when components and figures are not adjusted in a region and if exists space on the left/right in another region.
- **Figures and Components:** With the size reduction, figures and components can become unreadable. Therefore some rules must carefully be analyzed before being applied. They are:
 - Modification: modify the appearance of the components or figures;
 - Replacement: replace for some similar figure or component in case that the browser does not have support;
 - Removal: it removes the object or component in case that device does not have support to show it, as for example, figures on the cellular.

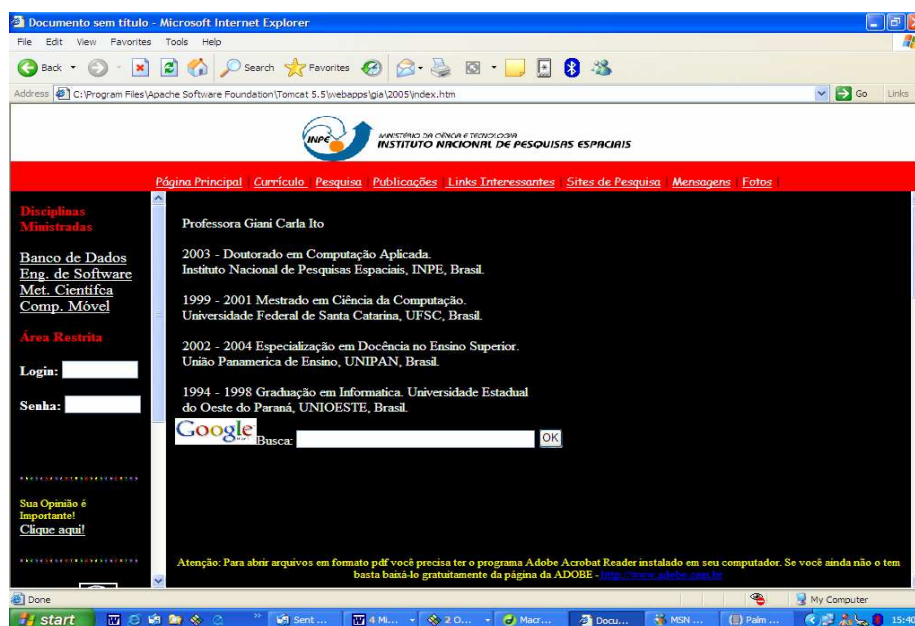


Figure 2. Original Site

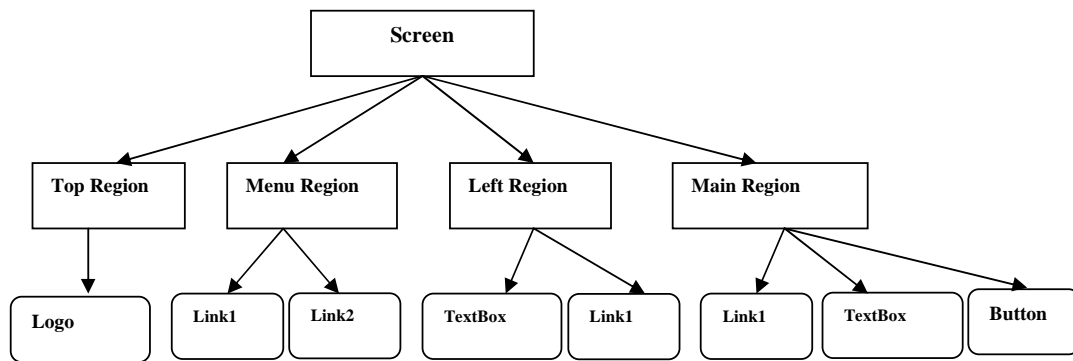


Figure 3. Hierarchical Site

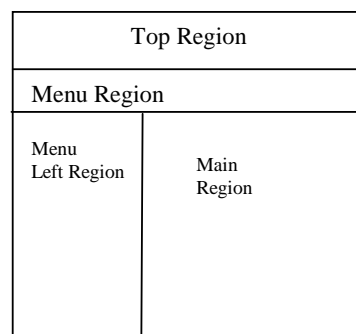


Figure 4. Site Divided in Regions

5. RELATED WORK & POSSIBLE SOLUTIONS

A servlet was developed to test the device recognition. Its function is to integrate the mobile device profile and a Web application. For these test it was used the server Apache Web TomCat 5.0, the Palm simulator 6.0.1 Cobalt and the Web browser 3.0.

Profiles had been stored in a local repository and the integration was proportionate by Delivery Context Library

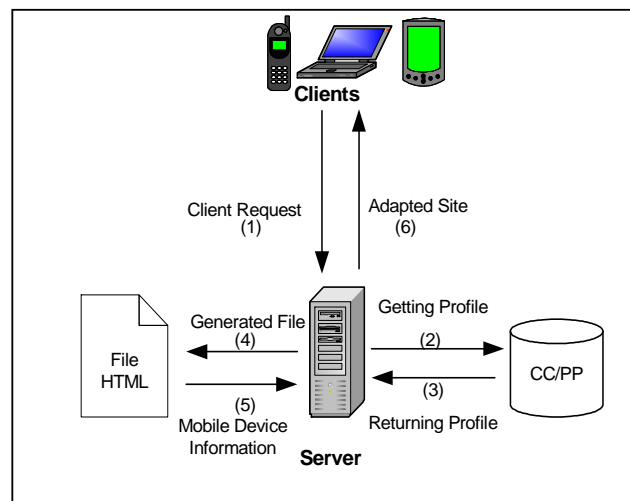


Figure 5. Acquiring Profile Device

(DELI). This library was developed by Hewlett Packward (HP) which allows the servlet to deal with devices requests that possess CC/PP profiles, Butler (2002). A CC/PP profile is composed by the platform hardware, the software and the browser components with their respective attributes. These attributes have characteristics as Resolution, Collection and Type. Each one has a value.

For demonstration, the first test presented the emulator profile detected in a HTML site, as illustrated on figure 6. Using a local profile repository, the tests present the speed advantage in which they are accessed.

The CC/PP profile can be observed on figure 5, which shows the archive generated content by the servlet, according to the client request. This figure also shows the hardware component in <http://www.wapforum.org/profiles/UAPROF/ccppschem-20010430#FramesCapable>, that contains the attribute in the address <http://www.wapforum.org/profiles/UAPROF/ccppschem-20010430#FramesCapable> informing if the browser used by the client supports frames.

Other attributes indicate the browser fabricant and if the same supports images, tables, the language type, the screen size and diverse characteristics that will be of great importance for the interfaces adaptation process in mobile devices.

Device Profile

Component	Attribute	Resolution	Collection	Type	Value
http://www.wapforum.org/profiles/UAPROF/ccppschem-20010430#SoftwarePlatform	http://www.wapforum.org/profiles/UAPROF/ccppschem-20010430#CcppAccept-Encoding	Override	Bag	Any	[base64]
http://www.wapforum.org/profiles/UAPROF/ccppschem-20010430#SoftwarePlatform	http://www.wapforum.org/profiles/UAPROF/ccppschem-20010430#CcppAccept	Override	Bag	Any	[image/gif, image/x- bitmap, image/jpeg, image/png, application/vnd.ms- powerpoint, application/vnd.ms- excel, application/msword, text/html]
http://www.wapforum.org/profiles/UAPROF/ccppschem-20010430#SoftwarePlatform	http://www.wapforum.org/profiles/UAPROF/ccppschem-20010430#CcppAccept-Charset	Override	Bag	Any	[US-ASCII, ISO-8859-1, UTF-8, ISO-10646-UCS- 2]
http://www.wapforum.org/profiles/UAPROF/ccppschem-20010430#SoftwarePlatform	http://www.wapforum.org/profiles/UAPROF/ccppschem-20010430#AcceptDownloadableSoftware	Override	Simple	Any	[No]
http://www.wapforum.org/profiles/UAPROF/ccppschem-20010430#BrowserUA	http://www.wapforum.org/profiles/UAPROF/ccppschem-20010430#TablesCapable	Override	Simple	Any	[Yes]
http://www.wapforum.org/profiles/UAPROF/ccppschem-20010430#BrowserUA	http://www.wapforum.org/profiles/UAPROF/ccppschem-20010430#FramesCapable	Override	Simple	Any	[Yes]
http://www.wapforum.org/profiles/UAPROF/ccppschem-20010430#BrowserUA	http://www.wapforum.org/profiles/UAPROF/ccppschem-20010430#BrowserName	Override	Simple	Any	[Microsoft]
http://www.wapforum.org/profiles/UAPROF/ccppschem-20010430#HardwarePlatform	http://www.wapforum.org/profiles/UAPROF/ccppschem-20010430#ImageCapable	Override	Simple	Any	[Yes]

Figure 6 - CC/PP Profile of Pocket PC

Based on profiles captured for a personal computer and a PDA a second test was taken using a site for desktop, as figure 1 shows. The site was elaborated using 4 frames, text figures, boxes, links and a button. When accessing the same site in the simulator, the complete layout is only visible with the use of horizontal scrolling bar and in the majority of the cases it is difficult to visualize what is shown on the mobile device screen, see figure 7 (a).

Some navigation problems can be perceived, as for example, the left menu presents itself extremely reduced, so it needs the horizontal scrolling bar for reading as the main menu. As the logo was not reduced it is not possible to visualize it totally. On figure 7 (b), the visualization with 50% of zoom was used, this resource was proportionate by the browser. However the reading of the content becomes an onerous process due to the size of the source.



Figure 7. (a) Palm Site (b) Site with reduced layout (c) Adapted Site

The figure 1 site was adjusted to the Palm simulator size of the screen, as figure 7 (c) shows. The mobile device interface is better visualized if programming the components of vertical form in contrast to the desktop which presents a horizontal layout. The CC/PP profile was used informing if the browser supports or not images, frames and the screen size.

The same site was accessed through a cellular simulator as showed on figures 8 (a), (b) e (c). The initial screen presents four links, one for each frame, where the user can choose which of them desires to have access. The top screen illustrated on figure 8 (b) sample the logo was practically unreadable. The left menu can be visualized on figure 8(c).

The biggest difficulties found on the site navigation in the cellular emulator were the division of cellular screens. The figures that own texts became practically unreadable. In this case the site did not present the return options, turning more difficult the navigation process.

According to the use of components after the tests accomplishment it be concluded that:

- **Scrolling Bar:** the use of the horizontal scrolling bar in only one direction must be limited. The use of the vertical scrolling bar in PDAs is indicated to facilitate the reading of the site content. It should be avoided the horizontal bar due to the difficulty of content visualization.
- **Colors:** avoid the use of many colors in the same site. The screens are small and the mixture of diverse colors can confuse the user.
- **Images:** Use small, simple images, and without text. With the reduction size process the texts can be totally unreadable;
- **Navigation:** It must kept one link for the main menu in all the sites to facilitate the navigation, however the user will always have to type the site address in each navigation;
- **Tables:** Not to use very long or bigger than screen size tables;
- **Texts:** In small screens the use of texts is more indicated than graphical in headings because they are easier to be adapted and visualized. The page titles should be short. It is indicated not to use more than three types of letters, avoiding to pollute the interface layout;
- **Usability:** In the mobile information essential devices must be shown on the first screen, therefore the faster the user finds what he desires, more usability the site will have. To the end of all the process of adaptation carried out for the architecture, the functionality will have to remain unbroken. This will be possible due to the use of the described restrictions in section 4 together with the techniques of the layout exhibition as follows:

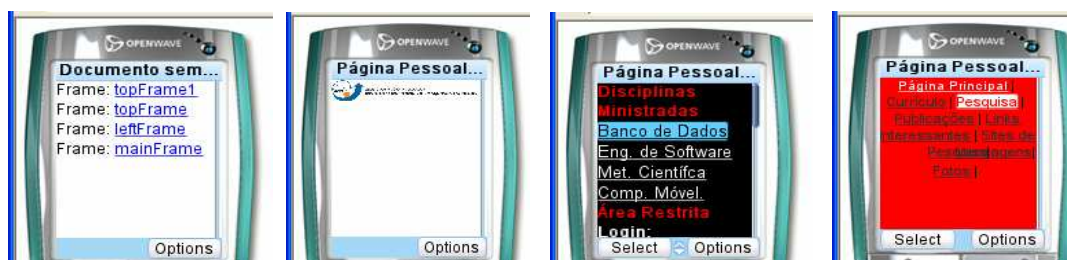


Figura 8. (a) Frames

(b) Logotipo

(c) Menu Esquerda

(d) Menu Superior

5.1. Layout

Two forms of content exhibition will be adopted in GAI architecture to turn possible the visualization either in a personal computer as in mobile devices: reduced layout and original layout.

- **Original Layout:** On this form of content presentation the original layout will be shown by complete, or either will be shown the site as it was projected. It can be problematic for the amount of information that must be visualized in the mobile devices, causing the lack of application usability, navigation difficulty and the content visualization, therefore it will be used for devices with bigger screens, as for example the PDAs.
- **Reduced Layout:** This form of exhibition diminishes the amount of information in the site sending only the relevant parts. All the sites are presented one after another in only one column, without the necessity of the horizontal bar clod roller. This way they could be visualized depending on the HTML site structure or based on the user preferences. A method to be adopted on this approach is the priorities one. On design time the developer can determine which regions will have to be shown, which one is more excellent, adding to the code the priority level of each region, in case the layout is reduced. This form of exhibition will be used for cellular devices.

6. CONCLUSIONS AND FUTURE WORK

This paper presented architecture for Adaptive Interfaces Generation for mobile devices based on the MVC model. This work is on development probation. The results gotten on the first stage referring to the mobile devices identification had been presented and the CC/PP characteristics for an application that provides the mobility resources, as well as, the accomplished tests had been shown with mobile devices simulators and emulators which are presenting the interface visualization on them.

Future work on this area has to include implementation of GAI architecture, providing an adaptation process on execution time, to the type of mobile device considering the class of devices, the platform type, and aspects of the user context among other factors. This way, with the devices identification using the CC/PP, associated with the user interfaces adaptation in mobile devices, it is desired to contribute for a bigger usability of them in the web environment.

7. REFERENCES

- Butler, M. H. (2002) DELI: A Delivery Context Library for CC/PP and UAProf. <http://www.hpl.hp.com/personal/marbut/DeliUserGuideWEB.htm>. August
- Coninx, Karin; Luyten, Kris; Vandervelpen, Chris; (2003) Dygimes: Dynamically Generating Interfaces for Mobile Computing Devices and Embedded Systems. <http://research.edm.luc.ac.be/kris/research>. March
- Hanumansetty, G. Reena. (2004) Model Based Approach for Context Aware and Adaptive User Interface Generation. Master's Thesis, Virginia Polytechnic Institute and State University, Falls Church, Virginia, USA.
- Menkhaus, Guido. (2002) Adaptive User Interface Generation in a Mobile Computing Environment. PhD Thesis, University of Salzburg, Austria
- Sendin, Montserrat; LORES, Jesus. (2004) Plasticity in Mobile Devices: a Dichotomic and Semantic View. AH 2004: 3rd International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems. Eindhoven. Holland. Proceedings del congress, page. 58-67, ISSN: 0926-4515.
- Sila, O.; Swick, R. (1999) Resource Description Framework (RDF) Model and Syntax Specification,

<http://www.w3.org/1999/status/REC-rdf-syntax-19990222/status>. January

Vermeulen, Jo. (2005) Widget Set Independent Layout Management For UIML, Masters thesis.
http://jozilla.be/data/publications/mscthesis-jo_vermeulen-text.pdf. February.