Journal of Mobile Multimedia, Vol. 3, No. 2 (2007) 179–196
© Rinton Press

A CONSTRAINT-BASED APPROACH TO DYNAMICALLY ADAPT MULTIMEDIA INTERFACES

JOSÉ M. OLIVEIRA[a]
INESC Porto / Faculdade de Economia, Universidade do Porto
Rua Dr. Roberto Frias, n. 378, 4200-465 Porto, Portugal
jmo@inescporto.pt

EURICO M. CARRAPATOSO
INESC Porto / Faculdade de Engenharia, Universidade do Porto
Rua Dr. Roberto Frias, n. 378, 4200-465 Porto, Portugal
emc@inescporto.pt

Received January 2, 2007
Revised May 29, 2007

The concept of the Virtual Home Environment (VHE) was introduced in mobile networks focusing on the issues surrounding the support of non-standard services for roaming users. This paper presents a proposal to solve the problem of the dynamic adaptation of multimedia services provided by a telecommunications operator in the context of the VHE concept. The paper discusses how the development of multimedia services can be facilitated in this context, how these services can be provided in a heterogeneous world of networks and terminals and how they can be used in a personalized and adaptable way.

The paper defines a solution for the adaptation of multimedia services based on the real-time generation of user interfaces conditioned by the user context. The solution is mainly characterized by the approach used for resolving the existing dependencies among user interface variables, which is based on the constraints theory, and by the mechanism for acquiring the user context information, which uses the Parlay/OSA interfaces.

Keywords: multimedia adaptation, Parlay/OSA, mobile environments, context gathering, SMIL
Communicated by: I. Ibrahim, J. Ng & C. Leung

1 Introduction

Mobile networks in general and particularly the Universal Mobile Telecommunications System (UMTS) has brought to the service provision context a new concept, known as Virtual Home Environment (VHE), that advocates services totally independent of network and terminal technologies [1]. The VHE concept gains particular importance in the provision of next generation services, characterized not only by location based features, but also by context-aware features, multimedia contents and user mobility. The acceptability of new services will only be effective if the user has the possibility to access them anywhere, in any technological circumstances, even in roaming scenarios. This user requirement places multimedia service providers and operators under the significant challenge of being able to transform their services in order to adapt them to a great variety of delivery contexts.

[a]Rua Dr. Roberto Frias, n. 378, 4200-465 Porto, Portugal.

This paper presents a generic adaptation methodology suitable to adapt telecommunications services to different access mechanisms, connectivity capabilities and user preferences. The methodology was designed in the perspective of a traditional telecommunications operator, which controls the network provision, the services provision and its set of user subscriptions, but opens the possibility to external trusted service providers of providing their services through open middleware, such as Parlay/OSA APIs.

In the following section, we compare our approach for the dynamic adaptation of multimedia services with some related work. The adaptation methodology, introduced in section 3, is based mainly on two fundamental constructions: the *Multimedia Presentation Model* (*MModel*) and *media adapters*. The former, presented in section 4, enables a device independent specification of the user interface, including the modeling of time-based features. *Media adapters* basically enable the materialization of the user interface specification in the most suitable format for a particular user context.

To explore the advantages of the adaptation methodology to produce multimedia services adapted to the user context and preferences, the paper presents in section 5 a *media adapter* targeted to the SMIL language. In this section we discuss the most relevant implementation issues of the *SMIL Media Adapter* prototype, stressing the mechanism for dynamically gathering the user context information, the approach taken to automatically generate the user interface, and the SMIL code generation process.

In section 6 we evaluate the proposed adaptation methodology and draw some conclusions regarding the qualitative aspects of the multimedia adapter, using a case study service.

The paper finalizes reporting the main conclusions in section 7.

2 Related work

The work on the *MModel* can be seen as parallel to the definition of the User Interface Markup Language (UIML) [13], which is a declarative language, currently being standardized by OASIS, that derives its syntax from XML and enables the description of device-independent user interfaces. In contrast with UIML, the definition of the *MModel* did not have the ambition of being a standardized way to define user interfaces but only had the objective of defining a model simple enough to demonstrate the VHE concept through the adaptation methodology. Another main difference between the *MModel* and UIML has to do with the target services. While UIML is more oriented toward Web services, the *MModel* was designed to define user interfaces of telecommunications services, provided in the context of telecommunications operators. In addition, the *MModel* should also give support to multimedia features, an important requirement for the current telecommunications services, not specifically covered by UIML.

In recent years, some research projects have been dedicated to the development of multimedia presentation systems following the constraint-based approach. *Cuypers* [18] is a research prototype system for the generation of Web-based multimedia presentations. Our work was greatly inspired by this system mainly because the system is design to operate in the context of a client/server architecture and also because it is mostly targeted to the Web environment. The core of the *Cuypers* system receives a semantic description of the multimedia document as its input and sends the generated presentation to the server, for further delivery to the client. The generation engine firstly produce a high-level semantic description of the presentation. Using a set of transformation rules (provided by the designer), the semantic structure is transformed into a specification based on *communicative devices*, which are abstract constructs that specify how the information should be conveyed to the user. The layout associated with the communicative device is specified at a high level using a set of *qualitative constraints*. The qualitative constraints are then transformed into *quantitative constraints* to define the exact positions of the various media items in the presentation. The last step corresponds to the encoding of the presentation in a format suitable for the player of the target device.

The integration of context information in telecommunications services and the development of context-sensitive applications have grown enormously in the present decade, mainly due to the crescent mobility of users and to the research activities on ubiquitous and pervasive computing. The SmartRoutaari [9] is a sound example of a context-sensitive system. SmartRotuaari is operational at the city center of Oulu, in Northern Finland, and comprises of a wireless multi access network (WLan, GPRS and EDGE), a middleware architecture for service provisioning, a Web portal with content provider interface and a collection of functional context-aware mobile multimedia services. The contextual information gathered by the SmartRoutaari system include the time, the location, the weather, the user preferences and the presence status. In contrast with our approach, the SmartRoutaari solution for gathering user context information is proprietary and has no facilities for the seamless expansion with independent third party services. This approach is not in line with the current standardization activities on telecommunications service provision, namely the 3GPP IMS standardization effort [5], which promotes the use of open APIs, such as Parlay/OSA, for opening operator networks to trusted third parties.

3 A methodology for the adaptation of telecommunications services

The adaptation methodology presented in this section is based on the *MModel*, an abstract construction that holds an XML description of the user interface, including the complete list of available user interactions. The objective is to create a mediator (the *Adaptation System*), between the service and the user, to manage all user interactions with the service. The *MModel* is a device independent model, which allows the capture of the essential service semantic and at the same time allows its adaptation to the current device characteristics, without losing the main service message to be passed to the user [10].

According to the actual service session parameters, the conversion of the *MModel* to a specific format and the adaptation of service contents to the right media type is achieved through specific implementations of the *media adapter* entity.

3.1 Initial access management

This section describes how the *adaptation system* decides and sets the most suitable *media adapter* for a specific user terminal connection. Each terminal connection is analyzed in terms of protocol by the *Protocol Handler*, which routes the call to the appropriate *Access Listener*. The *Protocol Handler* can be seen as a set of software callback routines, specific for different underlying connection technologies, that are waiting for call notifications (see figure 1).

Each *Access Listener* establishes the appropriate criteria used to choose the *Media Adapter* that will be associated with a connection. The criteria followed by the *Access Listeners* are strictly related with the protocol used by the terminal and define the format of information that the *Media Adapter* should produce. Taking into account the access protocol and the aspects concerning the user context, the *Adapter Chooser* decides which is the best *Media Adapter* available from the *Media Adapters Repository*. An instance of the chosen *Media*

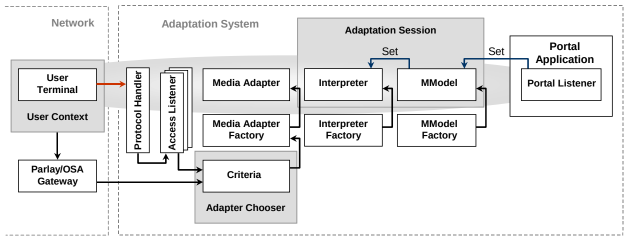


Fig. 1. Initial access management.

Adapter is then created and is immediately associated with the current adaptation session. Each *Adaptation Session* is composed by two objects: the *MModel*, which can be seen as the representation of a service on the *adaptation system* side; and the *Interpreter*, which can be seen as a mediator between the *Media Adapter* and the *Service Listener* (at this stage, the *Portal Listener*).

3.2 Service adaptation cycle

The user, after accessing his service provider, interacts with the portal application and chooses to use a service that he has previously subscribed. This interaction is transmitted to the *Portal Listener*, which triggers the creation of a new service session manager on the *adaptation system* side. This service session manager communicates with the service provider that offers the chosen service so that a service session manager on the service provider side and the associated listener can be also created (see figure 1).

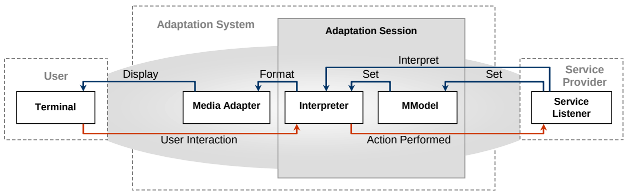


Fig. 2. User interaction and information display management.

The service session manager on the *adaptation system* side is also responsible for triggering the creation of the *Media Adapter*, the *Interpreter* and the *MModel* object instances, which will enable the service adaptation to the user context. The *Service Listener* is responsible for filling the *MModel* object with the user interface specification. This specification is usually divided into several parts, corresponding to the several views of the user interface.

The service should describe its user interface (following the *MModel* approach) reaching an equilibrium between the capabilities of the terminals that will most probably access the service and the message the service intends to pass. From this point of view, the user interface description that is sent to the *MModel* should always contain all the necessary information so that the service can be presented in the terminal with the best capabilities and in the best network conditions. If these are not the conditions of the context where the user is, the active *Media Adapter* should process the *MModel* in order to discard the information that cannot be presented (e.g., video information on WAP or DTMF terminals) and should process the information that may be conveyed given the current conditions (e.g., translations of text information to voice messages to be played on DTMF telephones), requiring the usage of another *Media Adapter* to perform this media content adaptation.

With the *MModel* set on the *adaptation system*, the service adaptation process starts, following a cycle approach (see figure 2). In each step, the adaptation is only carried out over one of the *MModel* parts. For that, the *Service Listener* sets the focus on the *MModel* part that should be adapted before it requests the Interpreter to interpret this part.

When the user receives a new interface in the terminal device, he has the opportunity to interact with it, leading the service to evolve to a new state. Each user interaction is characterized by the possible inputs introduced by the user and the order to go to a new state, typically represented by the reference of the interaction element. The *Interpreter* is the object that communicates the interaction description to the *Service Listener*. Using a parser, the *Service Listener* decodes the interaction and routes it to the right method of the service logic, which will process it and trigger the appropriate internal actions to answer the user interaction. Independently of the service, these changes in the service state will lead to an update of the *MModel* object, integrating the changes in the service state. The *Service Listener* sends a new user interface specification to the *MModel* or, in a situation where the user interface maintains a state similar to the previous one, the *Service Listener* only requests the change of some elements of the *MModel*. In both cases, the *Service Listener* should set the focus on the *MModel* part that it wants to be active.

4 The multimedia presentation model

The use of the *MModel* for the specification of the user interface follows a recursive approach. Each user interface component, even the simplest ones, should be represented by an `interfaceElement` and the entire user interface is, by itself, an `interfaceElement` (in this case, the root element of the *MModel* object). The `interfaceElement` should be either specialized into a `groupingElement` or a `simpleElement`, indicating that the element will be a holder of other interface components or will be an atomic interface component.

The *MModel* structure is depicted in figure 3, which only represents the relations between the various tags available. The *panel* elements have the particular role of being the elements over which the adaptation methodology will act, adapting their child elements and presenting them as a user interface view in the user terminal. There should be a special (and unique) *panel* element in the *MModel* document that is seen as the starting point of the user interface, being the first user interface part to be rendered in the user terminal. Thus, a service should structure its user interface into a sequence of *panels*, each one specifying the interface corresponding to a specific state of the service logic.

To facilitate the way service authors give a spatial/temporal layout structure to the user interface, the *panel* element is specialized into three elements. The `horizontalPanel` and the `verticalPanel`, which display their children elements, respectively, one before the other and one above the other. This is particularly beneficial for visual rendering terminals, where these elements can be used in a nested way giving a table-like aspect to the interface. However, they can also be used in other terminal types (e.g., voice terminals), following a specific rendering approach, such as displaying all the interface components from left to right and from top to bottom.

The `movingPanel` was introduced with the objective of adding time support to the *MModel*. This element has time attributes that give it the possibility of being displayed starting at a

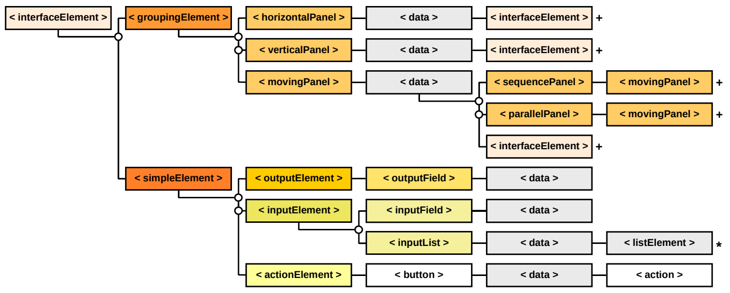


Fig. 3. User interface element types.

specific time and also of being active during a certain period of time. The movingPanel can be specialized into three element types: the sequencePanel, the parallelPanel and the interfaceElement. Both the *sequence* and the *parallel* panels are defined as a list of movingPanel elements. However, while in the parallelPanel these movingPanel elements should be played all starting at the same time, in the sequencePanel they should be played in sequence, following the order in which they appear in the *MModel* definition.

The element responsible for holding multimedia information items (e.g., text, image, video or audio), to be displayed in the user terminal, is the outputElement, which can be specialized into the outputField element. For enabling the user to input information while he uses the service, the *MModel* defines inputElement, which can be specialized into two element types: inputField and inputList. inputField is typically used when the service author wants to have an interface component which can receive the user input (e.g., a text box, a check box or an audio recorder). inputList contains a list of alternative options for being chosen by the user.

The element responsible for enabling the user to interact with the service is the actionElement, which can be specialized into the button element. Each button element is associated with an action element, which specifies the service listener that should be called when the user presses the associated button. This is an important feature in the adaptation methodology context, since the action element is the one that establishes a bridge between the user interface and the service logic.

The data element, whose tag structure is presented in figure 4, plays two roles in the *MModel*. On one hand, it works as a common description framework for presentable user interface components. On the other hand, a data element works also as a holder for the content information that is displayed in the user terminal (text and references to images and streams).

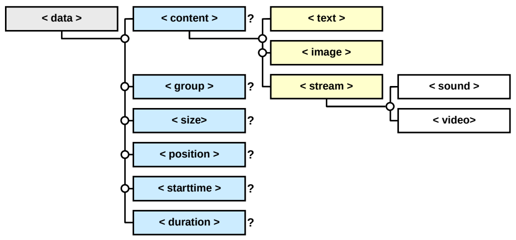


Fig. 4. data element tag structure.

5 A multimedia adapter

This section discusses the major features and the most important implementation aspects of a *media adapter* prototype specifically designed to enable the introduction of multimedia contents and to support time-based features in telecommunications services targeted for the VHE context [11].

The objective was to specify and develop a *media adapter* that should produce a final form user interface from an input interface specification defined using the *MModel* format. In what concerns the output format, the *media adapter* should produce information in an interactive format that would enable spatio-temporal relationships among the different interface components.

From the alternatives that may be considered as standardized multimedia formats, i.e., HyTime, MHEG, MPEG-4 and SMIL, we chose SMIL, a language defined by the World Wide Web consortium (W3C), which enables the specification of multimedia presentations for delivery over the Web [21]. The main reason behind this choice was the Web target of the SMIL language. Today, SMIL plays a role for synchronized hypermedia documents similar to the one played by HTML for hypertext documents [17]. We believe that the Web (provided over UMTS or WLan technologies) is the widest access means in the VHE context, and most of the terminal types may already access services provided through the Web. Besides that, the SMIL language structure was constructed following the XML syntax, which allows an easy mapping with the *MModel*.

Other factor with a major impact in the *media adapter* design was the approach taken for the dynamic generation of adaptable user interfaces according to the user context. The problem of dynamically generating user interfaces for multimedia services is mostly dependent on factors internal to the presentation design process itself, such as the presentation structure, the style or the content, and on external factors directly related with the user context, such as the network bandwidth conditions, the time available to see the presentation, the device characteristics where the presentation will be displayed, the cost associated to the access to the presentation or the user's personal preferences.

The adaptation methodology proposed by us prescribes the definition of the user interface structure as the initial point of the adaptation process, corresponding to the instantiation of the *MModel* object. Then, the interface structure is maintained stable during the different methodology phases, having no influence on the dynamic change of the user interface during the service session lifetime. The external factors mentioned above have a much more active role in the user interface dynamic generation, even influencing the style and the contents of the interface.

Among the alternative generation approaches usually used by multimedia presentation systems, the one based on constraints seemed to be the more natural way of coping with external factors in a dynamic way. The use of constraints is particularly appropriate to manage the spatial and temporal arrangements of the media items that are part of the interface, since the problem of establishing a final decision concerning the interface spatial and temporal layouts only involves finite and discrete variable domains.

5.1 User context gathering process

The notion of *context* and *context awareness*, from the point of view of mobile or ubiquitous computing, first appeared in the mid-nineties associated with the work of Schilit et al. [16].

Here, context is associated with the constant changes occurring in three execution environment levels: computing, user and physical environments. Since in mobile computing, the user context information is very dynamic by nature, the capability to dynamically detect this information at the beginning of each adaptation cycle is decisive to achieve acceptable results with the adaptation methodology.

The *SMIL Media Adapter* follows the approach of using the Parlay/OSA APIs [12] to obtain user context related information directly from the network. Figure 5 indicates which Parlay/OSA APIs are used in the user context gathering process. In the following paragraphs we detail the role of those APIs in obtaining the desired context information.

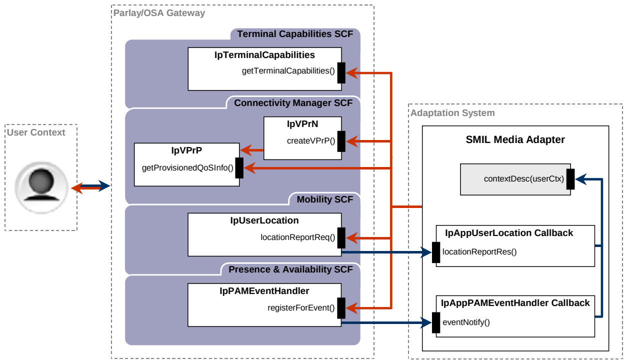


Fig. 5. User context gathering process using Parlay/OSA APIs.

Terminal Capabilities SCF: it enables an application to retrieve the terminal capabilities of a specific terminal, using the synchronous method `getTerminalCapabilities()` of the *IpTerminalCapabilities* interface (represented in figure 5). The application has to provide the terminal identity as the input to this method and the result indicates whether or not the terminal capabilities are available in the network and, if they are, the terminal capabilities will be returned. This information, if available, is returned as a CC/PP profile [20].

Mobility SCF: it provides a user location service, which allows applications to obtain the geographical location and the status (e.g., reachable or busy) of fixed, mobile and IP based telephony users. The application can interactively request a location report concerning one or several users invoking the `locationReportReq()` method of the *IpUserLocation* interface. This method is asynchronous, being the responses and reports handled by the *IpAppUserLocation* callback implemented by the *SMIL Media Adapter*.

Connectivity Manager SCF: the Parlay/OSA capability of allowing applications to constantly and efficiently obtain QoS monitoring information concerning the network conditions of user connections to service providers is still very limited [23]. Although the Connectivity Manager SCF provides this functionality, it was designed with the intention of providing QoS management on an enterprise scale and not for individual users. An application can obtain up-to-date network connection information (such as delay, loss, jitter and exceeded load parameters) through the method `getProvisionedQoSInfo()` of the *IpVPrP* interface and adapt itself in line with this information. The *IpVPrP* interface acts over a Virtual Provisioned Pipe (VPrP) service offered by the provider network to the enterprise network, which is a type of virtual leased line with pre-established QoS levels.

Presence & Availability Management (PAM) SCF: it offers a synchronous and an asynchronous approach to obtaining presence and availability information. Figure 5 depicts the asynchronous one, where an application registers its interest in receiving notifications concerning changes in the presence and availability states of a specific user, using the `registerForEvent()` method provided by the *IpPAMEventHandler* interface. The application indicates its callback interface (*IpAppPAMEventHandler*) that handles the notifications using the `eventNotify()` method. The synchronous approach is provided through the *IpPAMIdentityPresence* and *IpPAMAvailability* interfaces, which respectively provides the `getIdentityPresence()` and `getAvailability()` methods, which enable a direct access to presence and availability information.

The callback interfaces used to manage the asynchronous calls to the Parlay/OSA gateway invoke the `contextDesc()` method of the *SMIL Media Adapter* when they receive notifications from the gateway.

5.2 Constraints theory usage

The main idea behind the constraint-based approach for the automatic generation of multimedia presentations is to use a constraint solver system to determine one (preferably the best) solution for a set of variables that are interrelated by constraints. The *SMIL Media Adapter* internally uses the constraints theory to establish dependencies between user interface variables according to the user context and the preferences information gathered. The *SMIL Media Adapter* not only generates those constraints, but also interacts with an external constraint solver system to apply them.

The constraint solver system chosen to be used in the context of the *SMIL Media Adapter* was the ECL$^{\text{i}}$PS$^{\text{e}}$ constraint logic programming system [3], which not only offers a Java interface, but also the possibility to define application-dependent constraints, which is a very useful feature for multimedia applications, where the relations between the different interveners cannot be easily specified using typical numerical domains constraints. In addiction, ECL$^{\text{i}}$PS$^{\text{e}}$ supports the backtracking and unification features of logic programming, combining them with the domain reduction properties of constraint programming, resulting in what is usually called a Constraint Logic Programming (CLP) system. The use of the backtracking mechanism gives a dynamic characteristic to the specification of constraints. Alternative constraints can be specified when an initial constraint causes a failure, preventing the application from crashing or not providing any information.

The communication between the ECL$^{\text{i}}$PS$^{\text{e}}$ system and the *SMIL Media Adapter* is made through a queue mechanism provided by the ECL$^{\text{i}}$PS$^{\text{e}}$/Java interface (see figure 6). Two information queues are defined (`ToEclipseQueue` and `FromEclipseQueue`), which allow the *SMIL Media Adapter* to respectively send to and retrieve information from the ECL$^{\text{i}}$PS$^{\text{e}}$.

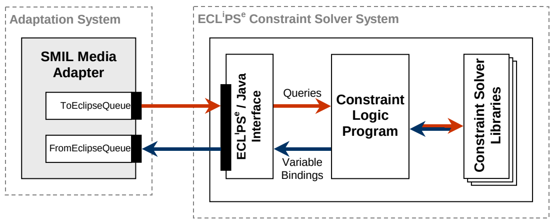


Fig. 6. Communication with the constraints solver system.

The *SMIL Media Adapter* firstly establishes the constraints that apply to the present situation. The inputs for this task are the *MModel* focused part and the user context description. The adapter parses both XML strings and extracts, from the first one, the characteristics of the media items that belong to the user interface and, from the second, the context information that impacts the constraints generation process.

Some media types, such as text and images, do not have an inherent notion of time in their specification. However, the constraint logic program defines for each media item, even for those that do not have time characteristics, a starting and an ending time variables. These media items can either inherit durations from other media items that have inherent durations when they play in parallel, such as presenting an image while an audio object plays, or have an imposed fix duration, assigned by the constraint solver system, following some predefined heuristic.

When the internal ECL^{i}PS^{e} Constraint Solver Libraries find a solution for the constraint problem, the *SMIL Media Adapter* extracts it from the `FromEclipseQueue` and updates the *MModel* focused part, namely with the spatial and temporal values found for each media item.

The qualitative constraint model followed by the *SMIL Media Adapter* for the basic spatio/temporal relations between media items is based on the Allen's relations [2], which are well described in [7]. Applying constraint handling rules, these qualitative constraints are translated into numeric constraints that can be solved by constraint-based systems built-in libraries.

However, those basic spatio/temporal relations are not sufficient to model some macro-coordination scale relations that should be defined in VHE service provision environments. The application of constraints motivated by VHE related issues to the space and time variables of a multimedia user interface is a way to guarantee that the interface will fit in a maximum screen size and with a maximum time duration, which are the two main requirements posed by multimedia services provided in VHE environments.

For example, if the *MModel* establishes a group of pictures to be displayed horizontally at the same time in a single screen, in a given situation the screen dimensions could prevent the display of all the pictures at the same time. This problem can be solved using the bookshelf approach [15], which displays the images from left to right along rows going from the top to the bottom of the screen. When there are too many images to be displayed in one screen, they are distributed among multiple screen displays, maintaining the rest of the interface the remaining information. The direct access to each of these displays is provided by hyperlinks.

Other problem happens when a particular picture has dimensions that exceed the screen dimensions. In this case, a possibility is to “augment” the screen dimensions, adding a scroll bar to the user interface. A better solution for this problem, mainly applicable to mobile devices that do not support scroll bars, is the resizing of the picture to a dimension that solves the constraint problem. The *SMIL Media Adapter* follows this adaptation strategy, relying also on more complex data transformations, such as image, audio and video format conversions, text to speech or speech to text translations.

Another strategy to enable the resolution of constraint problem inconsistencies was to associate each constraint with a priority weight. We follow a similar approach as Freeman et al. [6], where the constraints are ordered according to their relevance. Thus, a distinction is made between *required constraints*, which must be satisfied and, for that, are associated with a higher-level priority, and *preferred constraints*, which usually are related with the user preferences information and are associated with a lower-level priority. Lower-level priority constraints can even be deleted from the constraint problem when they create inconsistencies.

5.3 SMIL generation process

SMIL is not a content media format. Instead, it defines multimedia presentations as structured compositions of autonomous media objects [4]. Although integrating multimedia contents is its main function, SMIL was developed to remain a fully declarative format (based on the XML language), rather than a procedural language, enabling a direct map between the *MModel* and SMIL.

The *SMIL Media Adapter* uses the XSL Transformations (XSLT) [8] of the Extensible Stylesheet Language (XSL), which is a language specified by the W3C consortium for defining XML documents transformations and presentations. XSLT allows the specification, in a standardized way, of functional transformations from one XML tree to another, which are specified in a document usually known as a style sheet. The *SMIL Media Adapter* defines one such style sheet, which establishes the transformation rules from the *MModel* XML tree to the SMIL XML tree. The use of a relatively simple approach for transforming the *MModel* specification into a concrete delivery format using a XSLT transformation is driven by the guarantee that the calculations for the visual layout and timing of the media items produces an interface *MModel* tree that is known to meet the current user context.

SMIL is a very rich and extensive language, containing presently 37 elements and 183 attributes described in a 500 pages specification. The *MModel* mapping does not naturally cover the entire SMIL language, since in this case we would fall into two completely equivalent languages (SMIL and *MModel*) and that approach is not in line with the *MModel* purpose. The main objective of the *MModel* is to enable the specification of multimedia interfaces, independently of the target format used to deliver them to the user terminals. Being SMIL one possible format, the proposed mapping should enable the creation of SMIL presentations that convey the main message intended by the interface author.

The SMIL XSL style sheet only defines template rules for the *MModel* elements that have an effective role in the user interface. The other elements are seen as containers of effective elements and do not affect the SMIL code produced by the *SMIL Media Adapter*. To these container *MModel* source nodes, the SMIL XSL style sheet applies the `<xsl:apply-templates/>` instruction without producing any SMIL code, proceeding the transformation process to the next nested source elements.

Currently, most of the time-based formats are presentation formats. The SMIL was built as a Web presentation language, which supports hyperlinks to different parts of the presentation or to any external destination URI. However, an important drawback of the SMIL language, specifically concerning its use as the basis for implementing user interfaces, is the lack of support for input processing (usually designated as forms processing in the Web world). Web forms are one of the key features in the Web, as they offer a way to send user data to a server. This is particularly important in the context of the service adaptation methodology proposed in section 3, where the user inputs and interactions with the service should be processed at the service side, determining the next service state in a service session.

The SMIL linking elements are not a suitable choice to manage user interactions over a SMIL interface, since they are interpreted directly by the SMIL player, at the client side, being completely hidden from the service provider, which is not desirable in a service provision context. The intrinsic lack of support for forms processing should not be seen as a real drawback of the SMIL language, since currently the focus of W3C specifications is to separate the forms processing from specific languages, creating a standardizing way of treating them. This standardization effort is materialized into the XForms standard [22], which may be considered as an autonomous module that contains forms semantic-related XML elements and attributes and should be integrated in different XML based languages, such as XHTML, SMIL and SVG, extending these languages.

	MModel		SMIL 2.0	
	Elements	Attributes	Elements	Attributes
Grouping Elements	<horizontalPanel> <verticalPanel>	id	<region>	id
	<data> <size>	width, height		width, height
	<position>	x, y		left, top
	<starttime>	value		begin
	<duration>	value		dur
	<sequencePanel> <parallelPanel>	id	<seq> <par>	id
	<data> <starttime>	value		begin
	<duration>	value		dur
Output Elements	<outputField>	id		id
	<data> <content> <text>	label	<text>	src="data:, "
	<image>	url	<img>	src
	<stream> <sound>	url	<audio>	src
	<video>	url	<video>	src
	<size>	width, height		width, height
	<position>	x, y		left, top
	<starttime>	value		begin
	<duration>	value		dur

Table 1. *MModel* to SMIL mapping.

Taking these concerns about the forms processing into account, we decided to divide the *MModel* mapping process into two separate parts. The first mapping effort, presented in table 1, involves the processing of the *panel* and *output* elements of the *MModel*, which are transformed respectively into elements of the layout and timing modules and into elements of the media objects module of the SMIL language.

The second mapping effort, presented in table 2, involves the processing of the *input* and *action* elements, which are transformed into XForms elements. Besides those elements, identified in table 2, a SMIL interface that includes input and action elements should also define a form, using for that purpose the form definition process established by the XForms recommendation.

	MModel		XForms	
	Elements	Attributes	Elements	Attributes
Input Elements	<inputField>	id		
		type=text	<input>	id, ref
		type=encrypted	<secret>	id, ref
	<inputList>	id		
		multiple=true	<select>	id, ref
		multiple=false	<select1>	id, ref
	<listElement>	value	<item>	
			<label>value</label>	
			<value>value</value>	
Action Elements	<button>	id	<submit>	submission=id
	<data> <content> <text>	label	<label>	
	<action> <actionlistener>	reference	<submission>	id
				method="post"
				action=reference

Table 2. *MModel* to XForms mapping.

6 Experimentation and validation

As a proof of the concept we have implemented a prototype service with the objective of evaluating some of the most relevant research aspects proposed in this paper and to demonstrate the VHE concept itself. An overview of the *Customer Care* service in presented in figure 7, which clearly shows one of the VHE characteristics that is addressed by this service: the service ubiquitous access in terms of terminals and networks.

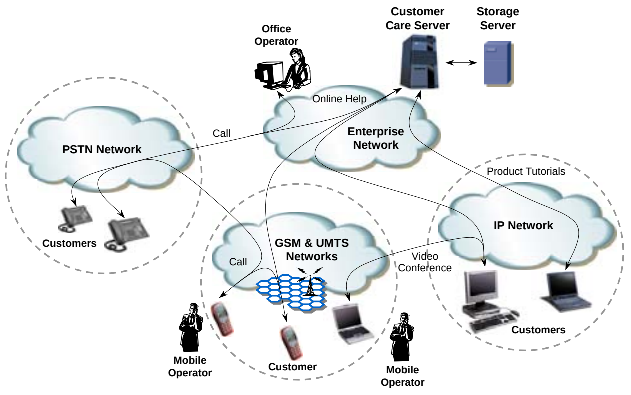


Fig. 7. Overview of the *Customer Care* service.

The *Customer Care* service suitability for evaluating the work presented here is directly related with the two features provided by the service: *interactive tutorials*, where the user interactively accesses information about products and services of a given supplier, organized as tutorials; and *online help*, where the user requests the establishment of an audio or videoconference with one of the available online operators. The multimedia flavor of the interactive tutorials enables the demonstration of the media adaptation to user preferences, through the use of profiles, and to different terminals and networks used to access the service. The multi-party characteristic of the online help feature enables the demonstration of network transparency, through the use of the Parlay/OSA Call Control API, and the demonstration of service adaptation to location information, through the use of the Parlay/OSA Location API.

6.1 Adaptation methodology evaluation

To evaluate the adaptation methodology we accessed the *Customer Care* service from three distinct terminals: a PC, a WAP terminal and a DTMF terminal.

For desktop and laptop PCs, the adaptation methodology applies to the *MModel* user interface specification a *media adapter* whose output language is HTML. In this case, all the information concerning the product tutorial is displayed without any change: information such as text, images, movie clips or audio is integrated in the HTML pages, following the predefined *MModel* user interface structure.

For WAP terminals, the application of a WML target *media adapter* produces WML pages, adapting the interface to a much more top-down interface style. In this case, no dynamic media (video or audio) is integrated in the user interface. For WAP terminals capable of displaying images, the WML *media adapter* can request image transformations to other *media adapters*, in order to resize and/or decrease the colormap of the interface images. For terminals that cannot display images, the WML version is limited to text information. In this case, the service is responsible for providing alternative text to substitute images and moving media.

For a DTMF terminal, such as an IP phone, the *MModel* is interpreted and adapted to an audio menu (DTMF menu), managed by an Interactive Voice Response (IVR) server to which the user is connected through the Parlay/OSA API. In this case, as a slide consists of text and images, the *media adapter* discards the images and converts the text to audio objects, using a text to speech translator.

The impact of the user context information in the service adaptation process was tested with the online help feature of the *Customer Care* service, which makes use of some Parlay/OSA APIs to significantly improve the service behavior, from the user point of view [14]. From one side, the Parlay/OSA Mobility API was used for obtaining the location of the users and the operators. With this information, the service determines the nearest available operator. This choice enables the reduction of the connection cost, which is a sound advantage for the user. From the other side, the Parlay/OSA Call Control API was used for sending an SMS to the chosen operator, notifying him of the user request. In the end of the scenario, the service transparently establishes a call (audio or video) between the user and the operator, again using the Call Control API.

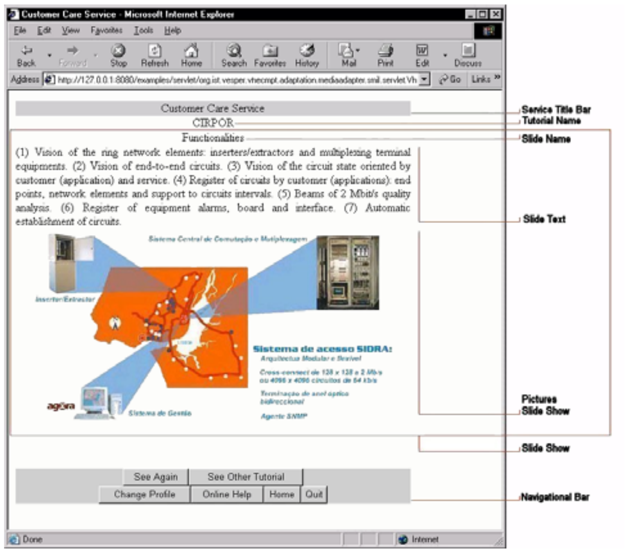
6.2 Multimedia adapter evaluation

This section evaluates the use of the constraints theory as a mechanism for supporting the dynamic adaptation of multimedia user interfaces in a VHE context. The obtained results are analyzed with respect to some broad qualitative parameters associated with the *media adapter*, which include consistency of the presented information, adaptation capability and presentation quality. A quantitative evaluation of the *SMIL Media Adapter* and, in particular, a performance evaluation falls in the scope of a constraint programming and solving strategy study, which was not within the objectives of our work.

The *Customer Care* service feature used for evaluating the *SMIL Media Adapter* was the product tutorials, which can be easily modeled as a multimedia presentation. Figure 8 illustrates the structure of a product tutorial presentation that should be displayed in ideal terminal and network conditions.

The still parts of the presentation are the service title bar, the tutorial name and the navigation bar, and they do not change during the presentation. The presentation moving parts are the slide show, which corresponds to the sequential display of each slide of information (the slide title and the slide text) and, inside each slide, the pictures slide show, which corresponds to a slide show of the set of pictures associated with a slide. According to different user context conditions, the above structure could have changed either its form, its content or both.

For experimenting and evaluating the *SMIL Media Adapter*, an essential requirement posed to the terminals was the possibility to display SMIL presentations. During the development, this requirement restricted the choice of the evaluation terminals to PCs and PDAs, which were the only terminals that allowed the installation of SMIL players. However, the variety of SMIL players was very restricted and the choice that proved to be the most appropriated for our work was the Internet Explorer, which implements the SMIL language profile designated XHTML+SMIL [19]. This choice had the advantages of being supported by a largely disseminated browser (even in PDAs) and of supporting the forms Web mechanism, which was an essential condition for implementing a service in line with the adaptation methodology defined in this paper. The main disadvantage was related with the different approach taken by the


Customer Care Service

CIRPOR

Functionalities

(1) Vision of the ring network elements: inserters/extractors and multiplexing terminal equipments. (2) Vision of end-to-end circuits. (3) Vision of the circuit state oriented by customer (application) and service. (4) Register of circuits by customer (applications): end points, network elements and support to circuits intervals. (5) Beams of 2 Mbit/s quality analysis. (6) Register of equipment alarms, board and interface. (7) Automatic establishment of circuits.

Customer Care Service - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print Edit Discuss
Address http://127.0.0.1:8080/examples/servlet/org.ist.vesper.vhecmpt.adaptation.mediaadapter.smil.servlet.Vh
Go Links
Sistema Central de Comutação e Multiplexagem
Inserter/Extractor
Sistema de acesso SIDRA:
Arquitectura Modular e flexível
Cross-connect de 128 x 128 a 2 Mb/s ou 4096 x 4096 circuitos de 64 kb/s
Terminação de anel óptico bidireccional
Agente SNMP
agora
Sistema de Gestão
See Again
See Other Tutorial
Change Profile
Online Help
Home
Quit
Done
Internet
Service Title Bar
Tutorial Name
Slide Name
Slide Text
Pictures Slide Show
Slide Show
Navigational Bar

Fig. 8. Product tutorial SMIL presentation structure.

XHTML+SMIL profile for managing the presentation spatial layout. This profile bases its layout functionality on the XHTML and Cascading Style Sheets (CSS) layout model and does not use the constructions defined by the layout module of the SMIL language. However, the presentations generated by both approaches are nearly equivalent, in what concerns spatial layout.

In SMIL, spatial media are placed within boxes with predefined sizes. The position of a media item is fixed in relation to the other media. Therefore, in our approach, determining the position of a media item is equivalent to finding its coordinates using a constraint solver. The data type that has the most important impact on the presentation spatial layout is the image data type. The *SMIL Media Adapter* uses a Java utility that extracts the image dimensions of each image that should be part of a presentation and integrates them in the constraint program.

The text data type is also modeled as a box, whose dimensions are calculated using an algorithm that considers the number of characters, the font size and the width of the box. Thus, the width of the text box is fixed by the presentation structure, by other media types and by the screen dimensions, varying only the text box height.

When the terminal was a laptop, the produced presentation had the spatial layout displayed in figure 8. In this case, no adaptation was performed, since the amount of text and the picture dimensions were all appropriate to fit in the screen dimensions. A different situation occurred when the terminal was a PDA, although the overall presentation structure shown in figure 8 was maintained. In this case, a primary adaptation operation was performed over text objects, adjusting the font size to an appropriate value, taking into account the small screen dimensions.

Then, because the width of some pictures of the tutorial is larger than the screen width, this fact forces the *SMIL Media Adapter* to adapt them, requesting the use of an image resize *media adapter*. The resize is performed if the percentage of the reduction does not go beyond a specified threshold, which is defined as a limit to preserve information consistency. If a specific picture needs a resize greater than the mentioned threshold to fit in the PDA screen dimensions, the *SMIL Media Adapter* takes the decision to drop it from the tutorial presentation.

In the product tutorial feature we assigned a start time and a duration to each slide displayable element. The start time of each element inside a slide is the same, except for the images that can be seen as a slide show inside the main slide show. We defined, as an input to the constraint program, the maximum duration of a presentation. This value is defined following an heuristic that takes into account the user profile and the subscription information. Different profile types, such as *user@home* or *user@office*, are usually associated with different network conditions, which can impact the presentation time and consequently the service cost. Also different contracted qualities of service provision (e.g., *gold* or *silver* qualities) surely impact the definition of the time-related constraints and even the content selection to be included in the final presentation.

Concerning the time dimension, the data that have more impact in the presentation global duration are audio clips and, in particular, speech. In a first approach, if the sum of the audio media items durations is lower than the maximum presentation duration, each slide will last a time equal to the correspondent audio clip duration. In this case, the duration of each picture inside a slide is equal to the slide duration divided by the number of pictures. To maintain the consistency of the presentation and guarantee a minimum presentation quality, we established constraints that force a minimum duration for each displayable element in the presentation, to avoid situations such as a text being displayed during only one second.

The constraints inconsistencies that occurred during our experiments were due to two different kinds of reasons. Firstly, when the durations associated with the audio clips are not consistent with the present constraints, the clips are eliminated from the presentation. This causes some inconsistency in the overall presentation, since some slides have audio and others do not. A solution for this problem is not easy to find and will always be service dependent.

The other source of constraints inconsistencies had to do with the media minimum duration constraints. These inconsistencies occurred in slides that had associated a large number of pictures, which led to a very short picture duration. In this case, the *SMIL Media Adapter* decides to drop a number of pictures from a slide, eliminating the constraints inconsistency.

7 Conclusions

The work presented in this paper is a proposal to solve the problem of the dynamic adaptation of multimedia services in the context of the VHE concept. We presented a generic adaptation methodology suitable for the adaptation of telecommunications services provided by an operator in a VHE context. The methodology follows the *single source, multiple deliveries* approach, using a conceptual model (the *MModel*) for the user interface specification. The *MModel* enables a device independent specification of multimedia user interfaces, promoting a clear separation between service structure and service presentation. This model should be seen as a powerful way for the rapid construction of context independent representations of multimedia user interfaces, which should then be materialized into the most suitable formats.

We presented the most important implementation aspects of a *media adapter* prototype (the *SMIL Media Adapter*) specifically designed to enable the introduction of multimedia contents and to support time-based features in telecommunications services targeted to be used in the VHE context. Being a standardized multimedia format for the Web, SMIL was chosen as the output format of the *media adapter* prototype described in this section. The choice of SMIL proved to be a significative advantage when it was necessary to define the mapping process between the *MModel* and SMIL, mainly due to the fact that both languages are based on XML.

We proposed the application of the constraints theory as the solution for defining multimedia user interface parameters adaptable to the service access conditions and the user context in the VHE, which include the physical properties of the user terminal, the network state, the user location and preferences. The *SMIL Media Adapter* relies on the use of open APIs, such as the Parlay/OSA APIs, to detect user contextual changes in mobile environments. These changes are then used to generate a set of constraints that the user interface should satisfy so that it may be properly displayed in the current conditions. We concluded that the usage of the constraints theory, as the solution for the dynamic generation of multimedia user interfaces, represents a successful example of the application of a well known computer science theory to the engineering field, namely to the telecommunications service area.

Acknowledgements

The work presented here was partially done in the framework of the "VESPER" and "OPIUM" projects, funded by the European Commission under contracts IST-1999-10825 and IST-2001-36063, respectively.

References

1. 3GPP. Service Aspects; The Virtual Home Environment. Technical Specification 3GPP TS 22.121 v5.3.0, Services and System Aspects Group, March 2002.
2. James F. Allen. Maintaining Knowledge about Temporal Intervals. *Communications of the ACM*, 26(11):832–843, November 1983.
3. Krzysztof Apt and Mark Wallace. *Constraint Logic Programming using ECLiPSe*. Cambridge University Press, 2006.
4. Dick C. A. Bulterman. SMIL 2.0, Part 1: Overview, Concepts and Structure. *IEEE Multimedia*, 8(4):82–88, October-December 2001.
5. Gonzalo Camarillo and Miguel-Angel García-Martín. *The 3G IP Multimedia Subsystem: Merging the Internet and the Cellular Worlds*. John Wiley & Sons, 2006.
6. Bjorn N. Freeman-Brenson, John Maloney, and Alan Borning. An incremental constraint solver. *Communications of the ACM*, 33(1):54–63, January 1990.
7. Joost Geurts. Constraints for Multimedia Presentation Generation. Master's thesis, Universiteit van Amsterdam, January 2002.
8. Michael Kay. XSL Transformations (XSLT) - Version 2.0. Proposed Recommendation PR-xslt20-20061121, W3C, November 2006.
9. Timo Ojala, Jani Korhonen, Markus Aittola, Mark Ollila, Timo Koivumäki, Jaana Tähtinen, and Heikki Karjaluoto. SmartRotuaari - Context-Aware Mobile Multimedia Services. In *Proceedings of the* 2^{nd} *International Conference on Mobile and Ubiquitous Multimedia*, Norrköping, Sweden, December 10-12, 2003.
10. José Oliveira, Renato Roque, Eurico Carrapatoso, Hans Portschy, Dániel Hoványi, and Imre Berenyi. Mobile Multimedia in VESPER Virtual Home Environment. In *Proceedings of the IEEE International Conference on Multimedia and Expo*, Lausanne, Switzerland, August 26-29 2002.
11. José Oliveira, Renato Roque, Manuel Dinis, and Eurico Carrapatoso. Provision of Mobile Multimedia over UMTS Middleware Platforms. In *Proceedings of the IST Mobile and Wireless Communications Summit*, Aveiro, Portugal, June 15-18 2003.

12. Parlay Group. Parlay Specifications, 2003.
13. Constantinos Phanouriou. *UIML: A Device-Independent User Interface Markup Language*. PhD thesis, Virginia Polytechnic Institute and State University, September 26 2000.
14. Renato Roque, José Oliveira, Filipe Pinto, Manuel Dinis, and Eurico Carrapatoso. 3G Service Provision Experience Over UMTS Middleware Platforms. In *Proceedings of the EURESCOM Summit - Evolution of Broadband Services*, Heidelberg, Germany, September 29 - October 2 2003.
15. Lloyd Rutledge, Jim Davis, Jacco van Ossenbruggen, and Lynda Hardman. Inter-dimensional Hypermedia Communicative Devices for Rhetorical Structure. In *Proceedings of the International Conference on Multimedia Modeling*, pages 89–105, Nagano, Japan, November 13-15 2000.
16. Bill Schilit, Norman Adams, and Roy Want. Context-Aware Computing Applications. In *Proceedings of the 1st IEEE Workshop on Mobile Computing Systems and Applications*, Santa Cruz, CA, US, December 1994.
17. Jacco van Ossenbruggen. *Processing Structured Hypermedia: A Matter of Style*. PhD thesis, Vrije Universiteit, April 2001.
18. Jacco van Ossenbruggen, Joost Geurts, Frank Cornelissen, Lynda Hardman, and Lloyd Rutledge. Towards Second and Third Generation Web-Based Multimedia. In *Proceedings of the 10th International World Wide Web Conference*, pages 479–488, Hong Kong, May 1-5 2001.
19. W3C. XHTML+SMIL Profile. Note NOTE-XHTMLplusSMIL-20020131, Synchronized Multimedia Working Group, January 2002.
20. W3C. Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies. Recommendation REC-CCPP-struct-vocab-20040115, Device Independence Working Group, January 2004.
21. W3C. Synchronized Multimedia Integration Language (SMIL 2.1) Specification. Recommendation REC-SMIL2-20051213, Synchronized Multimedia Working Group, December 2005.
22. W3C. XForms 1.0. Recommendation REC-xforms-20060314, XForms Working Group, March 2006.
23. Alvin Yew, Christos Bohoris, Antonio Liotta, and George Pavlou. Quality of Service Management for the Virtual Home Environment. In *Proceedings of the 12th International Workshop on Distributed Systems: Operations and Management (DSOM'01)*, Nancy, France, October 15-17 2001.