

Dynamic Generation of SMIL-Based Multimedia Interfaces

José Manuel Oliveira

INESC Porto, Faculdade de Economia, Universidade do Porto

Eurico Carrapatoso

INESC Porto, Faculdade de Engenharia, Universidade do Porto

Rua Dr. Roberto Frias, 378, 4200-465 Porto, Portugal

Email: {jmo, emc}@inescporto.pt

Abstract—The current heterogeneous world of networks and terminals combined with the generalized mobility of users pose telecommunications operators and service providers the challenge of dynamically adapt the services they provide. This paper presents a proposal to solve the problem of the adaptation of multimedia services in mobile contexts. The paper combines context-awareness techniques with user interface modeling and description to dynamically adapt telecommunications services to user resources, in terms of terminal and network conditions. 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. The experiments and tests carried out with these techniques demonstrate a general improvement of the adaptation of multimedia services in mobile environments, in comparison to systems that do not dynamically integrate the user context information in the adaptation process.

Index Terms—multimedia adaptation, dynamic generation, mobile environments, context gathering, SMIL

I. INTRODUCTION

Currently, the evolution of telecommunications is mainly driven by the convergence trend of three traditionally separate worlds: Internet, broadcast and mobile. This trend is supported by some standardization efforts, being the IP Multimedia Subsystem (IMS) [1], promoted by the 3rd Generation Partnership Project (3GPP), the most important one. IMS can be seen as a unified architecture that supports a wide range of multimedia IP-based services to be provided to users across different access networks and device types, with consistent look and feel, and ease of use.

The acceptance of new services in this context 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 under the significant challenge of being able to transform their services in order to adapt them to a great variety of delivery contexts. This need

for multimedia service adaptation in mobile environments constituted the main motivation for the research work presented in this paper.

The paper defines a generic adaptation methodology targeted to adapt multimedia services provided in the context of a telecommunications operator. The methodology is based mainly on two fundamental constructions: the *Multimedia Presentation Model (MModel)* and *Media Adapters*. The former enables a device independent specification of the user interface, while *Media adapters* basically enable the materialization of the user interface specification in the most suitable format for a particular user context. The proposed methodology follows the approach of dynamically integrating the user context information, which is by nature very changeable in mobile environments, in order to achieve consistent results in the adaptation process. Following a key point of IMS, which is being an open-systems architecture, we base the approach for the context gathering process on the Parlay middleware, a set of open APIs that enable trusted service providers to directly access network resources, traditionally restricted exclusively to the network operator.

The paper is organized as follows. Section II compares our approach for the dynamic adaptation of multimedia services with some related work. Section III presents a generic adaptation methodology suitable to adapt telecommunications services to different access mechanisms, connectivity capabilities and user preferences. Section IV presents the most relevant implementation issues of a *Media Adapter* targeted to the Synchronized Multimedia Integration Language (SMIL). Section V evaluates the proposed adaptation methodology and draws some conclusions regarding the qualitative and quantitative aspects of the multimedia adapter, using a case study service. Section VI reports the paper main conclusions.

II. RELATED WORK

The problem of adapting multimedia services and presentations has received significant attention from the

research community in the last years. *Cuypers* [2] is a research prototype system for the generation of Web-based multimedia presentations following a constraint-based approach, which was the main inspiration for the constraints generation process of our adaptation methodology. A drawback of *Cuypers* is its focus on the delivery of SMIL content for desktop PCs, not exploring other final presentation formats and devices.

In [3] and [4], the problem of delivering multimedia content to different end device types, using a wide range of multimedia formats, is covered. However, the process used by these systems for acquiring the user context is not dynamic. Besides that, these two adaptation systems do not consider the possibility of media adaptation or transformation, adopting the approach of previously defining alternative media, similar to the SMIL `switch` tag.

The SmartRoutaari [5] is another example of a context-sensitive system. SmartRoutaari is operational at the city center of Oulu, in Northern Finland, and comprises 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 time, location, weather, user preferences and 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 [6], which promotes the use of open APIs, such as Parlay/OSA, for opening operator networks to trusted third parties.

The use of an abstract multimedia model as the basis for the adaptation process has been a usual approach in this area. Examples of such models are UIML [7], ZyX [8] and DISL [9], which is an extension of a subset of UIML to enhance the support for interaction modeling. A common characteristic of these models is their ambition to be as generic as possible. UIML is even a standardized language, defined within the OASIS consortium. In contrast with those models, 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 adaptation methodology.

A particular characteristic of the *MModel* has to do with its target services. The *MModel* was designed to define user interfaces of telecommunications services, provided in the context of telecommunications operators. In addition, the *MModel* also gives support to multimedia features, an important requirement for current telecommunications services.

The use of SMIL in the telecommunications field increased significantly with the 3GPP adoption of the 3GPP SMIL Language Profile as the media synchroniza-

tion and presentation format for the Multimedia Message Service (MMS) [10]. However, the use of SMIL in the context of interactive services is still very limited. In [11], the authors propose several extensions to SMIL to cover input and output capabilities not currently available, such as location-based information, telephony, forms, scripting and tactile feedback. From these capabilities, the support for forms was crucial during the development of the work presented in this paper. To add forms support to SMIL, we followed the World Wide Web Consortium (W3C) approach of integrating XForms with SMIL through the definition of a joint mapping from *MModel* to these two languages.

Our work is also related with the recent advances in the area of accessing multimedia services in mobile contexts, through small devices [12], [13], [14]. In these works, the adaptation is usually done directly in the content, by adapting the media codification or by transcoding the media elements. Although the methodology we propose is not optimized for small devices, being the main focus on the capability to cover a large range of access devices and also to be open and scalable to support new future device types, the integration of those adaptation techniques in the *SMIL Media Adapter* had a significant impact on the results obtained (see Section V).

III. 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 *MModel* is a device independent model, which allows the capture of the essential service semantics and at the same time allows its adaptation to the current device characteristics, without losing the main service messages to be passed to the user [15].

The use of the *MModel* for the specification of the user interface follows a recursive approach. Each user interface component, even the simplest ones, must be represented by an `interfaceElement` and, at the same time, the entire user interface is, by itself, an `interfaceElement` (in this case, the root element of the *MModel* object). The `interfaceElement` is 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.

A service is modeled as a set of *panels*, corresponding each one to the user interface of a specific state of the service logic (see Figure 1). These *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 is a special *panel* element in the *MModel* document that is seen as the starting point of the user interface, being the service first user interface part to be rendered in the user terminal.

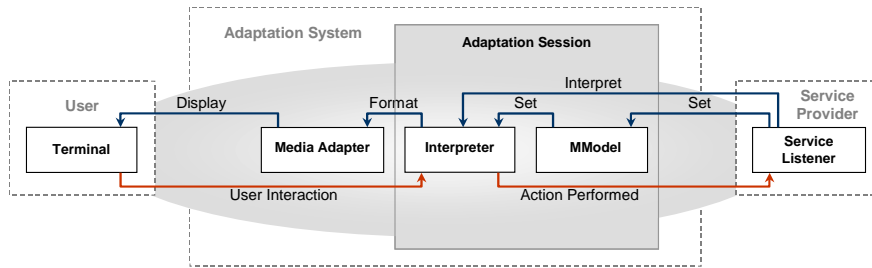


Figure 2. User interaction and information display management.

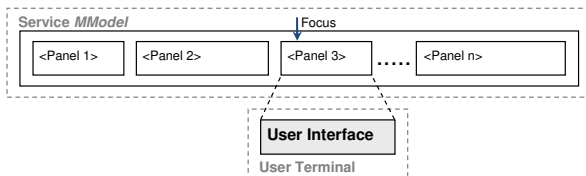


Figure 1. Structure of an *MModel* representation of a service.

Each of these *panels* has a temporal and a spatial structure, build through the `horizontalPanel`, the `verticalPanel`, the `sequencePanel` and the `parallelPanel` elements. These interface structure elements hold active interface elements, such as the `outputField`, for multimedia information items (e.g., text, image, video or audio); the `inputField` and `inputList` for enabling the user to input information while he uses the service; and the `button` element, for enabling the user to interact with the service. 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, since the `action` element is the one that establishes a bridge between the user interface and the service logic.

A service author should design a service user interface (following the *MModel* approach) establishing a compromise between the capabilities of the terminals that will most probably access the service and the message the service aims 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 according to the used network. If these are not the conditions of the context where the user is, the *MModel* should be processed in order to discard the information that cannot be presented (e.g., video information on Wireless Application Protocol (WAP) or Dial Tone Multi Frequency (DTMF) terminals) and to transform 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).

A. Service adaptation cycle

The entity responsible for performing the adaptation of the *MModel* to a specific format (or media type) is the *Media Adapter*. The *Media Adapter* plays an important role at two distinct levels of service provisioning: at the

user interface realization, according to the access network and the terminal type used to access the service; and at the content provisioning from the service to the user terminal.

The *Adaptation System* decides and sets the most suitable *Media Adapter* for a specific user terminal connection according to a criteria strictly related with the protocol used by the terminal to access the service provider and with the user context parameters read at connection time. The decision on the best *Media Adapter* available from the *Media Adapters Repository* defines the format of information that should be produced from the *MModel* service representation.

Figure 2 illustrates the cyclic approach followed by the adaptation methodology to adapt the service user interface to the current user conditions. The user, having accessed his service provider, interacts with a portal application and chooses to use a service that he has previously subscribed to. An *Adaptation Session* is created in the *Adaptation System* to manage the entire adaptation process. 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*.

The *Service Listener* is responsible for filling the *MModel* object with the user interface specification. With the *MModel* set on the *Adaptation System*, the service adaptation process starts. In each step, the adaptation is carried out over only 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 go through this part (see Figure 1).

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 an 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. In addition, the *Service Listener* should set the focus

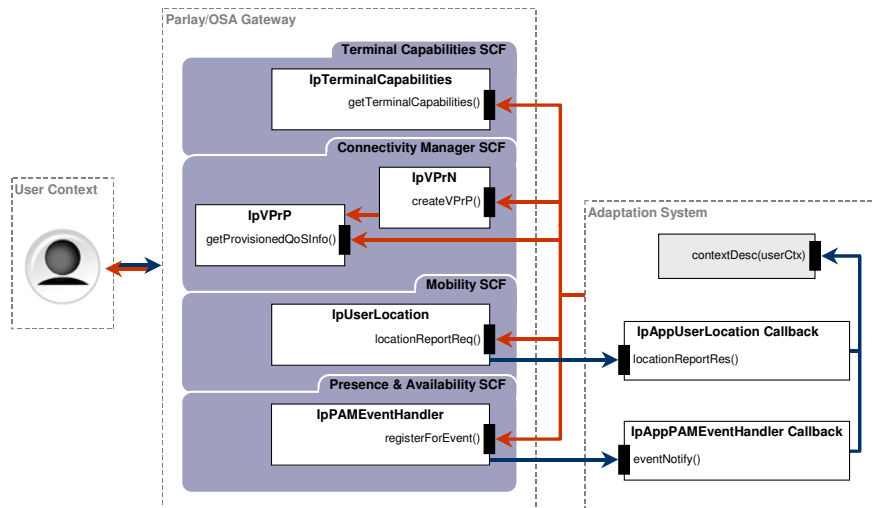


Figure 3. User context gathering process using Parlay/OSA APIs.

on the *MModel* part that it wants to be active.

B. 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. In the present work, we restrict the notion of context to computing and user execution environments.

The adaptation methodology follows the approach of using the Application Programming Interfaces (APIs) offered by Parlay/OSA [17] to obtain user context related information directly from the network. These Parlay/OSA APIs enable telecommunications network operators to open up their networks to third party service providers, in a secure and trusted way.

Figure 3 indicates which Parlay/OSA Service Capability Features (SCFs) are used in the user context gathering process. The *Terminal Capabilities* and the *Connectivity Manager* SCFs enable the gathering of computing environment information related with the terminal and the network connection used to access the service, while the *Mobility* and the *Presence & Availability Management* SCFs enable the gathering of user environment information related with the location and the status of the user. In the following paragraphs we detail the role of those SCFs in obtaining the desired context information.

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. 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 Composite Capability/Preference Profiles (CC/PP) profile [18].

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 *Adaptation System*.

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 limited to 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 provides presence and availability information. 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 also indicates its callback interface (*IpAppPAMEventHandler*) that handles the notifications using the `eventNotify()` method.

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

IV. A MULTIMEDIA ADAPTER

This section discusses the major features and the most important implementation aspects of a *Media Adapter* prototype specifically designed to handle multimedia contents and to support time-based features in telecommunications services targeted for mobile environments. 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.

Among 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 W3C, which enables the specification of multimedia presentations for delivery over the Web [19]. The main reason behind this choice was the Web targeting of the SMIL language. Today, SMIL plays a role for synchronized hypermedia documents similar to the one played by HTML for hypertext documents [20]. The Web (provided over UMTS or WLAN technologies) is already important in mobile contexts, 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*.

Another 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 results mostly from factors internal to the presentation design process itself, such as presentation structure, style or content, and on external factors directly related with the user conditions when he accesses and uses the service, generally designated by user context. Examples of external factors are the network bandwidth conditions, the time available to see the presentation, the device characteristics where the presentation will be displayed, the cost associated with 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, by the service designer, 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 various 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

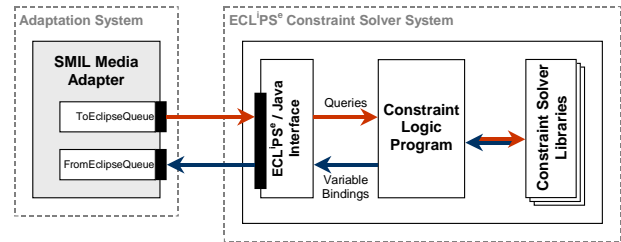


Figure 4. Communication with the constraints solver system.

dynamic generation, even influencing the style and the contents of the interface.

Among the alternative generation approaches usually used by multimedia presentation systems, which include the operational approach [21] and the constraints-based approach [2], the one based on constraints is the most suitable to cope 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.

A. 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 constraint solver system chosen to be used in the context of the *SMIL Media Adapter* was ECLiPS^e [22], 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 addition, ECLiPS^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 ECLiPS^e system and the *SMIL Media Adapter* is made through a queue mechanism provided by the ECLiPS^e/Java interface (see Figure 4). Two information queues are defined (`ToEclipseQueue` and `FromEclipseQueue`), which allow the *SMIL Media Adapter* to respectively send and retrieve information from the ECLiPS^e.

Figure 5 presents the algorithm based on the use of constraints technology in the *SMIL Media Adapter* to produce an updated *MModel* representation of the

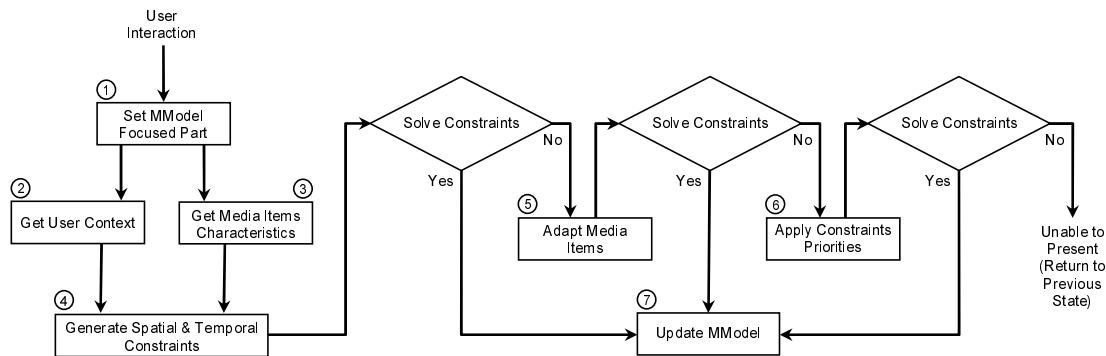


Figure 5. Algorithm to produce an updated *MModel*.

service. Following the user interaction in the user interface, the service listener sets the focused part of the *MModel* (Step 1). Then, using the Parlay/OSA APIs (see Section III-B), the *SMIL Media Adapter* obtains the user context description (Step 2) and extracts from the *MModel* focused part the characteristics of the media items that belong to the user interface (Step 3). These media items characteristics are the width and height of visual items and the duration of all media items.

Some media types, such as text and pictures, 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 inherit their durations from other media items that have inherent durations when they play in parallel, such as presenting a picture while an audio object plays, or have an imposed fixed duration, assigned by the constraint solver system, following a predefined heuristic related with the context where the user accesses the service (e.g., longer durations for fixed accesses and shorter durations for mobile accesses).

The information collected in steps 2 and 3 enables the *SMIL Media Adapter* to establish the spatial and temporal constraints that apply to the present user situation (Step 4), following the basic spatial and temporal constraints model detailed in Section IV-A.1).

When the internal ECLiPS^e Constraint Solver Libraries find a solution for the constraints problem, the *SMIL Media Adapter* extracts it from the `FromEclipseQueue` and updates the *MModel* focused part (Step 7), namely with the spatial and temporal values found for each media item. However, mainly in mobile environments where the terminals have restricted capabilities, the constraints problem may not have a solution. To solve these cases, additional constraints strategies are used by the *SMIL Media Adapter* for obtaining better results in the *MModel* adaptation to the current user conditions. These additional strategies are detailed in Section IV-A.2).

1) **Spatial and temporal constraints:** Visual media, such as images, text and video, are characterized by the two usual spatial x and y dimensions and also by the z dimension, which describes the layering order of overlapping items. Only continuous media, such as audio

and video, are intrinsically characterized by the temporal t dimension. However, when included in a presentation, all media items are assigned a position along the temporal axis and have also a duration, as said before.

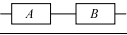
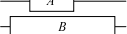

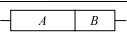
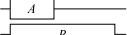


Being A and B two media items, the z dimension is often modeled using qualitative constraints given by the following three possible relations: A in front of B , A behind B and A beside B . These three qualitative constraints have a direct mapping onto quantitative constraints in the following way: $A < B$, $B > A$ and $A = B$.

The qualitative constraint model usually applied to the temporal dimension is based on the 13 interval relations specified by Allen [23] (see first column of Table I). The table presents the relations applied to the t dimension and only presents 7 relations since the first 6 have their corresponding inverse relation that can be inferred. The Allen’s model is also applied to the spatial x and y dimensions, being suitable for most of the applications.

In order to check if the above qualitative constraints do not violate some hard quantitative requirements, such as the target platform’s screen size or the maximum duration of the presentation, they should be mapped onto numeric constraints (quantitative constraints) so that they can be solved by constraint-based built-in libraries. Each of the qualitative Allen’s relations can, for each of the three dimensions x , y and t , be mapped onto one or more numeric constraints. The second column of Table I defines this mapping for the t dimension, using T_1 as the starting time of a media item and T_2 as the ending time. These numeric constraints are produced in Step 4 of the algorithm.

2) **Additional constraints strategies:** The basic spatio/temporal relations presented above are not sufficient to model some macro-coordination scale relations that should be defined in the mobile service provision 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 [24], which displays the pictures from left to right along rows going from the top to the bottom of the screen. When there are too many pictures to be displayed in one screen, they are distributed

TABLE I.
ALLEN'S RELATIONS AND THEIR TRANSLATION ONTO
QUANTITATIVE CONSTRAINTS.

Allen's relation	Quantitative constraints
A before B 	$T_2^A < T_1^B$
A during B 	$T_1^A > T_1^B, T_2^A < T_2^B$
A overlaps B 	$T_1^A < T_1^B, T_2^A > T_1^B, T_2^A < T_2^B$
A meets B 	$T_2^A = T_1^B$
A starts B 	$T_1^A = T_1^B, T_2^A - T_1^A < T_2^B - T_1^B$
A finishes B 	$T_2^A = T_2^B, T_2^A - T_1^A < T_2^B - T_1^B$
A equals B 	$T_1^A = T_1^B, T_2^A = T_2^B$

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. We integrate this approach in Step 4 of the algorithm.

Another problem happens when, for example, a particular picture has dimensions that exceed the screen. In this case, a possibility is to "augment" the screen dimensions, adding a scroll bar to the user interface. However, there are some mobile devices that do not support scroll bars. Before excluding this element from the presentation, a negotiation about the format can be tried in order to replace the element by another version. In this case, the resize of the picture to a dimension that solves the constraint problem is a possible solution. The *SMIL Media Adapter* follows this adaptation strategy (Step 5), relying also on more complex data transformations, such as picture, audio and video format conversions, text to speech or speech to text translations, performed by specific *Media Adapters* that are available in the *Adaptation Systems*.

Concerning the temporal dimension, we defined, as an input to the constraints program, the maximum duration of a presentation. This value is defined following a heuristic that takes into account the number of media items in the presentation, 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. Thus, long durations are defined for *user@office* profiles and for high quality subscriptions.

Another strategy to enable the resolution of constraint problem inconsistencies is to associate each constraint with a priority weight (Step 6). We follow a similar approach as Freeman et al. [25], 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 and are associated with a lower-level priority. The lower-level priority constraints

are deleted from the constraint problem when they create inconsistencies.

B. SMIL generation process

SMIL is not a content media format. Instead, it defines a multimedia presentation as structured composition of autonomous media objects. 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 mapping between the *MModel* and SMIL.

The *SMIL Media Adapter* uses the XSL Transformations (XSLT) [26] 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 such a 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 produce 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.

Currently, most of the time-based formats are presentation formats. SMIL was built as a Web presentation language, which supports hyperlinks to different parts of the presentation or to any external Uniform Resource Identifier (URI). However, an important characteristic 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 of 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 III, 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

TABLE II.
MModel MAPPING: (A) TO SMIL 2.0 LANGUAGE; (B) TO XFORMS LANGUAGE.

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

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

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 standardized way of treating them. This standardization effort is materialized into the XForms standard [27], which may be considered as an autonomous module that contains XML elements and attributes related with semantic aspects of forms. The XForms module should be integrated in different XML based languages, such as XHTML, SMIL and SVG, extending these languages.

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 II-(A), 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 II-(B), involves the processing of the *input* and *action* elements, which are transformed into XForms elements. Besides those elements, identified in Table II-(B), 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.

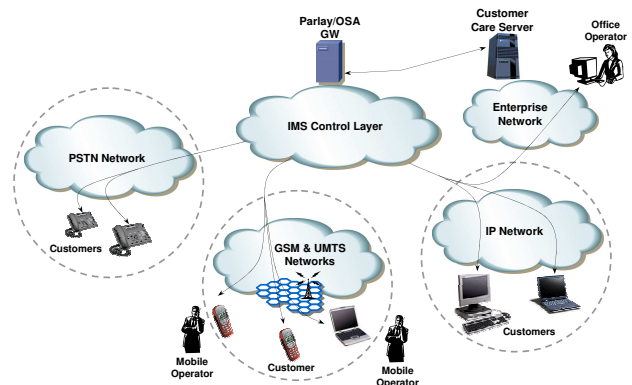


Figure 6. Overview of the *Customer Care* service.

V. EXPERIMENTATION AND VALIDATION

As a proof of the concept we have implemented a prototype service with the objective of evaluating some of the relevant research aspects proposed in this paper. The choice was a *Customer Care* service. An overview of this service is presented in Figure 6, which clearly shows one of adaptation methodology characteristics that is addressed by this service: the service ubiquitous access in terms of terminals and networks.

The *Customer Care* service suitability for evaluating the work presented here is directly related with the *interactive tutorials* feature provided by the service, where the user interactively accesses information about products and services of a given supplier, organized as tutorials. The multimedia flavor of the interactive tutorials enables, in a first place, the demonstration of the adaptation



Figure 7. SMIL presentation structure of the *Interactive tutorial* feature: (a) Displayed on a laptop. (b) Displayed on a PDA.

methodology by allowing the *Customer Care* service to be used through different terminals and networks, and then the evaluation of the constraints-based approach followed by the *SMIL Media Adapter*.

A. Adaptation methodology demonstration

To demonstrate the adaptation methodology we accessed the *Customer Care* service from four distinct terminals: a laptop, a PDA, a WAP terminal and a DTMF terminal. Figure 7 shows a screenshot of the *interactive tutorials* feature displayed in a laptop (on the left side) and in a PDA (on the right side). In both cases, the adaptation methodology applies to the *MModel* user interface specification a *Media Adapter* whose output language is the SMIL language. The user interface structure is the same for both terminals, although some media items were adapted in the PDA access, to cope with the screen limited dimensions.

Figure 8 shows the *Customer Care* service being accessed from a WAP terminal. The application of a Wireless Markup Language (WML) target *Media Adapter* produces WML pages, adapting the interface to a much more top-down style. In this case, no dynamic media (video or audio) is integrated in the user interface.

In 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. 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.

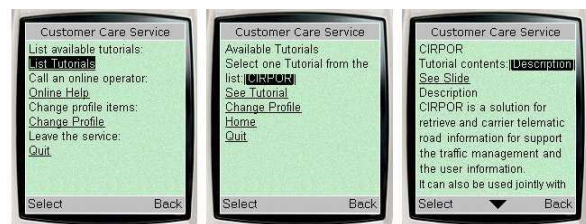


Figure 8. *Interactive tutorial* WAP presentation structure.

B. 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 heterogeneous contexts. Firstly, the *SMIL Media Adapter* is evaluated with respect to some broad qualitative parameters, which include the consistency of the presented information, the adaptation capability and the presentation quality. In addition, the impact of the amount of media items integrating the user interface in the *SMIL Media Adapter* behavior is analyzed. The tested parameters were the adapter average error rate and the initial presentation delay time.

The *interactive tutorials* feature of the *Customer Care* service can be easily seen and modeled as a multimedia presentation. The left side of Figure 7 illustrates the structure of an *interactive tutorial* presentation as it would 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

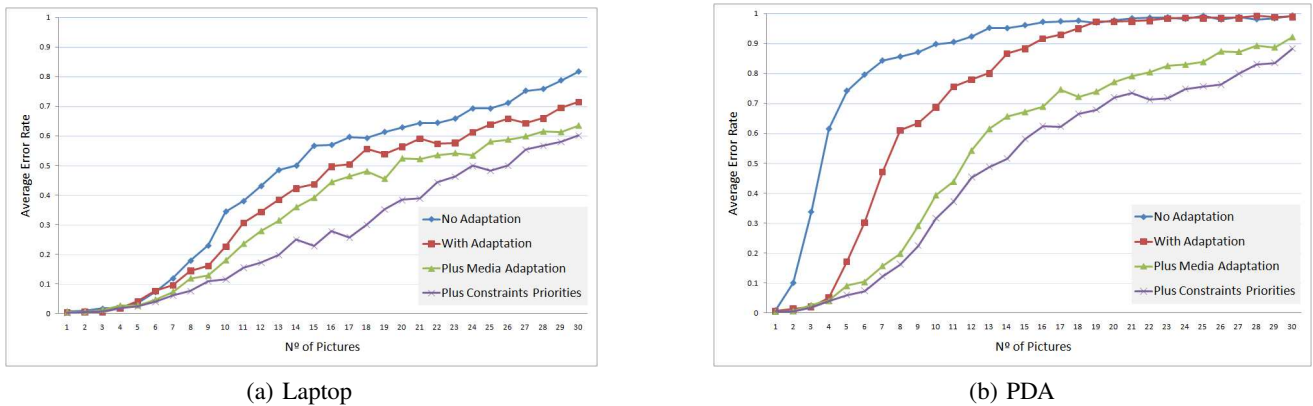


Figure 9. Presentation average error rate according to the number of pictures.

moving parts are the slide show, which corresponds to the sequential display of each slide (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 can change either in form, content or both.

We tested the *SMIL Media Adapter* in two terminals, a laptop and a PDA, both with the Internet Explorer browser as the SMIL player. The Internet Explorer implements the SMIL language profile designated by XHTML+SMIL. This choice had the advantages of being supported by a very disseminated browser (even in PDAs) and of supporting the Web forms 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 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 equivalent, in what concerns spatial layout.

In a PDA terminal, although the overall presentation structure shown in the left side of Figure 7 is maintained, the number of adaptation operations carried out by the *SMIL Media Adapter* is significantly higher than in large terminals. A primary adaptation operation is usually 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* (during Step 5 of the algorithm presented in Section IV-A). The resize is performed if the percentage of the reduction does not go beyond a specified threshold, usually 50% of the original size, 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, the *SMIL Media Adapter* takes the decision to drop it

from the tutorial presentation.

Concerning the temporal 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 temporal 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 minimum media duration constraints. These inconsistencies occurred in slides that had associated a large number of pictures, which led to a very short picture presentation.

In order to evaluate the efficiency of the *SMIL Media Adapter*, a series of tests have been carried out. The objective was to analyze the behavior of the *SMIL Media Adapter* when faced with an increase of media items in a presentation. We used the *interactive tutorials* feature of the *Customer Care* service, generating a tutorial that always contained the same contents except for the number of pictures of the slide show. We considered slide shows containing a single picture up to slide shows containing 30 pictures. For each number of pictures, 50 tests were performed. The pictures were randomly picked from a database containing around 600 pictures of various dimensions. The smallest picture had 120 × 90 pixels, having the biggest 1200 × 860 pixels.

Figure 9 compares the impact of the slide show number

of pictures on the *SMIL Media Adapter* error rate (i.e., the percentage of times the constraints problem does not have a solution) for four distinct levels of adaptation, closely related with the algorithm presented in Section IV-A:

- **no adaptation**, i.e., the *SMIL Media Adapter* only converts the tutorial *MModel* to SMIL code, without applying the constraints theory, and consequently without taking into account the user context. In this case, the *SMIL Media Adapter* only applies Step 1 of the constraints algorithm;
- **with adaptation**, i.e., the *SMIL Media Adapter* generates the basic temporal and spatial constraints (Step 4), taking into account the user context and the tutorial contents, and solves them;
- **plus media adaptation**, i.e., when Step 4 of the algorithm produces a set of constraints that has no solution, the *SMIL Media Adapter* applies the media adaptation step of the algorithm (Step 5);
- **plus constraints priorities**, i.e., when the media adaptation step still produces a set of unsolved constraints, the *SMIL Media Adapter* applies constraints priorities to the constraints problem (Step 6), enabling the deletion of low priority constraints.

The left side of Figure 9 shows the results when the tutorial is displayed in a laptop, while the right side presents the equivalent results considering a PDA as the accessing terminal. As expected, the *SMIL Media Adapter* performance decreases with the increase of the number of pictures in the slide show. This tendency is more notorious in smaller terminals, where the average error rate increases in the first half of the graphic. However, in both terminals, each of the above adaptation levels contributes to a better performance of the *SMIL Media Adapter*, having a significant impact in the PDA tests, where the application of the basic constraints and the media adaptation steps are responsible for the display of the tutorial in a large number of cases.

The importance of the proposed constraints algorithm in the *SMIL Media Adapter* performance is clear in the PDA graphic. Here we can see that, for a large number of pictures (above 20), the basic temporal and spacial constraints step (Step 4) does not introduce any difference in the adapter performance. For these cases, the two last algorithm steps (media adaptation and constraints priorities) offer much better results.

Using the same test ambient presented above, Table III presents the variation of the average initial delay presentation time and the maximum initial delay presentation time of the *SMIL Media Adapter*, according to the number of pictures in the tutorial slide show. Comparing the times measured in PDAs with the laptop times, they are significantly higher. This is mainly due to the media adaptation step that starts to be used intensively for slide shows with 10 or more pictures. However, this delay is due to the media adaptation operation itself, since when the *SMIL Media Adapter* decides to perform the media adaptation over a specific tutorial element, it can, in parallel, request the constraints problem resolution, running both processes

TABLE III.
INITIAL PRESENTATION DELAY TIMES ACCORDING TO THE NUMBER OF PICTURES AND THE TERMINAL TYPE.

Number of Pictures	Laptop		PDA	
	Mean Delay	Maximum Delay	Mean Delay	Maximum Delay
4	0.2 s	1.2 s	1 s	8 s
8	0.6 s	2.2 s	5 s	12 s
12	2 s	10 s	7 s	18 s
20	6 s	18 s	10 s	25 s

simultaneously, possibly in different machines.

VI. CONCLUSIONS

The work presented in this paper is a proposal to solve the problem of the dynamic adaptation of multimedia services in heterogeneous contexts. We presented a generic adaptation methodology suitable for the adaptation of telecommunications services according to different conditions of access terminal and network. The methodology uses 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.

We presented the most important implementation aspects of a *Media Adapter* prototype (the *SMIL Media Adapter*). Being a standardized multimedia format for the Web, SMIL was chosen as the output format of the *Media Adapter* prototype. This choice proved to be a significant advantage when it was necessary to define the mapping 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 dynamically adaptable to the user context, 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, in real time, 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 contents may be properly displayed in the current conditions. This dynamic integration of the user context information in the adaptation process constitutes the main advantage of our approach, in comparison to similar adaptation systems. From the large number of tests done to evaluate the performance of the *SMIL Media Adapter*, we conclude that the proposed constraints algorithm can achieve significant improvements in the efficient adaptation of multimedia services targeted to mobile terminals.

ACKNOWLEDGMENT

The work presented here was partially carried out 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] V. Koukoulidis and M. Shah, "The IP multimedia domain: service architecture for the delivery of voice, data, and next generation multimedia applications," *Multimedia Tools and Applications*, vol. 28, no. 1, pp. 203–220, 2006.
- [2] J. van Ossenbruggen, J. Geurts, F. Cornelissen, L. Hardman, and L. Rutledge, "Towards Second and Third Generation Web-Based Multimedia," in *Proceedings of the 10th International World Wide Web Conference*, Hong Kong, May 1-5, 2001, pp. 479–488.
- [3] A. Scherp and S. Boll, "Paving the Last Mile for Multi-Channel Multimedia Presentation Generation," in *Proceedings of the 11th International Multimedia Modelling Conference*, Melbourne, Australia, January 12-14, 2005, pp. 190–197.
- [4] P. Bertolotti, O. Gaggi, and M. L. Sapino, "Dynamic context adaptation in multimedia documents," in *Proceedings of the 2006 ACM Symposium on Applied computing*, New York, NY, USA, 2006, pp. 1374–1379.
- [5] T. Ojala, J. Korhonen, M. Aittola, M. Ollila, T. Koivumäki, J. Tähtinen, and H. Karjaluo, "SmartRotuaari - Context-Aware Mobile Multimedia Services," in *Proceedings of the 2nd International Conference on Mobile and Ubiquitous Multimedia*, Norrköping, Sweden, December 10-12, 2003.
- [6] G. Camarillo and M.-A. García-Martín, *The 3G IP Multimedia Subsystem: Merging the Internet and the Cellular Worlds*. John Wiley & Sons, 2006.
- [7] C. Phanouriou, "UIML: A Device-Independent User Interface Markup Language," Ph.D. dissertation, Virginia Polytechnic Institute and State University, September 26, 2000.
- [8] S. Boll and W. Klas, "ZYX - A Multimedia Document Model for Reuse and Adaptation of Multimedia Content," *IEEE Transactions on Knowledge and Data Engineering*, vol. 13, no. 3, pp. 361–382, 2001.
- [9] W. Müller, R. Schäfer, and S. Bleul, "Interactive Multimodal User Interfaces for Mobile Devices," in *Proceedings of the 37th Hawaii International Conference on System Sciences*, Waikoloa, HI, USA, November 2004.
- [10] 3GPP, "Multimedia Messaging Service (MMS); Media Formats and Codecs (Release 7)," Services and System Aspects Group, Technical Specification 3GPP TS 26.140 v7.1.0, June, 2007.
- [11] K. Pihkala and P. Vuorimaa, "Nine Methods to Extend SMIL for Multimedia Applications," *Multimedia Tools and Applications*, vol. 28, no. 1, pp. 51–67, 2006.
- [12] Y. Hwang, J. Kim, and E. Seo, "Structure-Aware Web Transcoding for Mobile Devices," *IEEE Internet Computing*, vol. 07, no. 5, pp. 14–21, 2003.
- [13] W. Y. Lum and F. Lau, "User-Centric Adaptation of Structured Web Documents for Small Devices," in *Proceedings of the 19th International Conference on Advanced Information Networking and Applications*, March 28-30, 2005, pp. 507–512.
- [14] Y. Chen, X. Xie, W.-Y. Ma, and H.-J. Zhang, "Adapting Web Pages for Small-Screen Devices," *IEEE Internet Computing*, vol. 9, no. 1, pp. 50–56, 2005.
- [15] J. Oliveira, R. Roque, E. Carrapatoso, H. Portschy, D. Hoványi, and I. 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.
- [16] B. Schilit, N. Adams, and R. 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] Parlay Group, "Parlay Specifications," 2003. [Online]. Available: <http://www.parlay.org/en/specifications/>
- [18] W3C, "Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies," Device Independence Working Group, Recommendation REC-CCPP-struct-vocab-20040115, Jan. 2004. [Online]. Available: <http://www.w3.org/TR/2004/REC-CCPP-struct-vocab-20040115/>
- [19] —, "Synchronized Multimedia Integration Language (SMIL 2.1) Specification," Synchronized Multimedia Working Group, Recommendation REC-SMIL2-20051213, December, 2005. [Online]. Available: <http://www.w3c.org/TR/2005/REC-SMIL2-20051213/>
- [20] J. van Ossenbruggen, "Processing Structured Hypermedia: A Matter of Style," Ph.D. dissertation, Vrije Universiteit, April, 2001.
- [21] J. Yu and Y. Xiang, "Hypermedia Presentation and Authoring System," in *Proceedings of the 6th International World Wide Web Conference*, California, USA, April 7-11 1997.
- [22] K. Apt and M. Wallace, *Constraint Logic Programming using ECLiPSe*. Cambridge University Press, 2006.
- [23] J. F. Allen, "Maintaining Knowledge about Temporal Intervals," *Communications of the ACM*, vol. 26, no. 11, pp. 832–843, November, 1983.
- [24] L. Rutledge, J. Davis, J. van Ossenbruggen, and L. Hardman, "Inter-dimensional Hypermedia Communicative Devices for Rhetorical Structure," in *Proceedings of the International Conference on Multimedia Modeling*, Nagano, Japan, November 13-15, 2000, pp. 89–105.
- [25] B. N. Freeman-Brenson, J. Maloney, and A. Borning, "An incremental constraint solver," *Communications of the ACM*, vol. 33, no. 1, pp. 54–63, January, 1990.
- [26] M. Kay, "XSL Transformations (XSLT) - Version 2.0," W3C, Proposed Recommendation PR-xslt20-20061121, November, 2006. [Online]. Available: <http://www.w3.org/TR/2006/PR-xslt20-20061121/>
- [27] W3C, "XForms 1.0," XForms Working Group, Recommendation REC-xforms-20060314, March, 2006. [Online]. Available: <http://www.w3.org/TR/2006/REC-xforms-20060314/>

José Manuel Oliveira received his PhD degree in engineering science from the University of Porto in 2005, his MS degree in electrical engineering and computer science from the University of Porto in 1996, and his BS degree in mathematics and computer science from the University of Porto in 1992.

He began his academic career, in 1995, at the Mathematics and Informatics Group of Faculty of Economics, University of Porto, where he is currently an Assistant Professor. He is also a researcher at the Telecommunications and Multimedia Group of INESC Porto since 1992. His current research interests include the dynamic adaptation of multimedia services and telecommunications services provision on mobile environments.

Eurico Carrapatoso received his PhD degree in information systems engineering, in 1987, from the University of Bradford, U.K and a degree in electrical engineering, in 1977, from Universidade do Porto. He began his academic career, in 1977, at Faculty of Engineering, University of Porto, where he is currently an Assistant Professor. He teaches simulation, computer networks, distributed systems, multimedia and digital signal processing.

He joined INESC Porto in 1987, where he leads a research group in the area of Advanced Telecommunication Services over Mobile Networks. His main research activities and interests are in simulation, high speed networks architectures and protocols, and advanced telecommunication services, particularly distributed multimedia applications and services.