

Context-aware design of adaptable multimodal documents

Augusto Celentano · Ombretta Gaggi

© Springer Science + Business Media, LLC 2006

Abstract In this paper we present a model and an adaptation architecture for context-aware multimodal documents. A compound virtual document describes the different ways in which multimodal information can be structured and presented. Physical features are associated to media instances, while properties describe the context. Concrete documents are instantiated from virtual documents by selecting and synchronizing proper media instances based on the user context: the situation, the environment, the device and the available communication resources. The relations between the context features and the media properties are described by a rule based system.

Keywords Hypermedia design · Adaptation · Rule system

1. Introduction

The growth of information service providers and a wide spread of features in network performance (e.g., Ethernet, Wi-Fi, GPRS), in user devices (e.g., desktop, notebook, PDA, smart phone), in user context (e.g., location, profile and environment variants), and in user interfaces (e.g., visual, auditory and embedded interfaces) make compelling demand of models and systems for documents able to adapt themselves to the user environment.

In this paper we present a model and an adaptation architecture for designing and instantiating context-aware multimodal documents, considering in a unified way media features, device, user and environment properties, through a rule-based definition of the relationships between the context and the document components.

A. Celentano (✉) · O. Gaggi
Dipartimento di Informatica, Università Ca' Foscari di Venezia,
Via Torino 155, 30172 Mestre VE, Italia
e-mail: auce@dsi.unive.it

O. Gaggi
Current address: Dipartimento di Matematica Pura e Applicata,
Università di Padova, Via G.B. Belzoni, 7, Padova, Italia
e-mail: gaggi@math.unipd.it

Adaptation and context-awareness are emerging features in several information technology areas, such as human computer interaction, applications and services; the two terms are tightly related, and sometimes are used as synonyms. However, they denote two distinct concepts. In the document domain, *adaptation* means the ability to provide different versions or different presentations of a document in order to suite the needs of a user or the features of a device; *context-awareness* means the capability of perceiving the user situation in its many aspects, and of adapting to it the document content and presentation.

Adaptation is the ultimate outcome of context-awareness, which is able to drive it without explicit user tuning operations. In the human-computing interaction (HCI) area, a further distinction is made between *adaptable* and *adaptive* systems [12]: in adaptable systems the user chooses among different behaviors, while adaptive systems provide automatic modification of the behavior according to the user needs.

Context-awareness goal is therefore the design of adaptive, rather than adaptable systems. However, in other fields, such as in the software engineering community, the term *adaptable* is used more frequently to cover both meanings, and we shall use it in this paper.

Adaptable documents are characterized by being multimedia, multichannel and multimodal. In the scope of this paper we'll assume the following meaning for such terms:

- a *multimedia* document is a time-varying document (sometimes called a *multimedia presentation*) containing several synchronized media items, with different degrees of user control, from simple VCR style commands to hypermedia navigation;
- a *multichannel* document can be delivered, in different formats and with different content versions, through different communication channels, such as wide band or GPRS networks, and different presentation devices, such as desktop screens or portable phone displays;
- a *multimodal* document content belongs to several communication codes, and is perceived by a human with different senses, in parallel or in alternative, such as vision, audio and touch (e.g., vibration).

The three properties above usually co-exist, at some extent: a multimodal document has a multimedia content, and is also multichannel since the different communication codes are delivered through different channels. In the following we'll use the term *multimodal document* to subsume also the multimedia and multichannel properties.

2. Related work

A lot of work has been done in recent years in the area of context-awareness and adaptation. The term context has been initially associated to the concept of location, but is far richer, as underlined by several works [14, 21, 22]. The context is defined by Dey as “any information that can be used to characterize the situation of an entity. An entity is a person, place or object that is considered relevant to the interaction between a user and an application” [9]. Such information includes several facets, related to the user situation (location and time, mobility, profile), to

the environment (indoor vs. outdoor, light vs. darkness, sound vs. silence), to the communication (wide vs. narrow band, plugged vs. wireless connection, continuous vs. discontinuous transmission), to the device (screen size and resolution, audio fidelity, input devices, user attention), etc. Some properties are relevant in specific application fields for deciding how to adapt a document; e.g., heavy traffic conditions prevent a driving user from receiving unanticipated visual messages on the navigator screen, while audio messages can be received because they are less distracting.

Adaptation has been studied in domains like mobile computing, Web applications, user interfaces and hypermedia [5, 7, 16, 23]. Several proposals exist for automatic adaptation of multimedia documents, but they are generally limited to some aspects of the context (e.g., location, device features, processing resources) and do not involve the whole process of document design and delivery.

Adaptation of multimedia documents to user location is typical in tourism and cultural heritage applications (see for example [1, 8, 19]). In such applications, adaptation consists in selecting from a repository the information relevant for the user location, sensed by sensors systems like GPS, GSM, infrared beams or other means. User motion can trigger the delivery of new information, which is presented according to a template which may also depend on the user device and on the environment (e.g., indoor vs. open space).

Content adaptation is performed to satisfy the compliance of a multimedia document with devices and resources. The content of the document is processed to fit the constraints imposed by the available resources, and the dynamic behavior of multimedia components is adapted to overcome delays and shifts due to resource fluctuations, according to suitable temporal models. In [32] a temporal model for multimedia presentations is discussed, which adapt themselves to a context described in terms of network bandwidth, device and available resources; the presentation behavior can vary within specified limits. Euzenat et al. [10] discuss how to constrain the temporal aspects of a multimedia document to device and network resources, by introducing a notion of semantics upon which two different concepts of adaptation (preserving or not preserving all the properties of the original document) are defined.

Adaptation is also performed by transforming an original multimedia information with operations which change physical properties like frame size and rate, audio sampling, color depth, in order to fit the device constraints and the network limitations. The transformation can be done “on the fly” at delivery time, or pre-processed alternative versions of the same information can be computed in advance and selected upon delivery. In both cases, the overall dynamic structure of the presented information does not change. Adaptation shares a number of features with the automatic generation of multimedia presentations, which is performed by evaluating a set of constraints about the presentation content and layout. Such systems (see for example [2, 28]) build different presentations with the same “conceptual” content by selecting different media files, semantically equivalent, and adapting them to the layout and to the resources.

More complex adaptation techniques, like the one discussed in this paper, introduce a higher level of adaptation which applies to the logical structure of the multimedia document, not only to its content, while preserving the meaning of the information delivered. In [4], the authors propose an adaptation system in which the alternatives evaluated during the adaptation process include media items

of different types, with different composition schemas. E.g., a video file can be replaced by a sequence of images and a separate audio comment, as long as they refer to the same meaning. Adaptation can also be done during the presentation playback, upon a context change.

In [11] adaptation is approached with a different viewpoint. Assuming that different combinations of media could convey the same meaning to the user, the authors present adaptation as the selection and combination of media which have the greatest cognitive impact on the user, according to psychology studies about the human perception.

The use of standard languages for describing documents and context related information, such as the ones based on XML, makes document adaptation easier due to a formal ground for transforming and validating documents along all the processing phases. SMIL [27] has been the first standard language devoted to the description of synchronized multimedia documents. It provides a simple mechanism for adapting parts of a document to a part of the playback context. The tag `<switch>` can be used to test a restricted set of features at the client side, such as the operating system used, the network bandwidth, the screen features, and to select consequently different components of the document. In [24], Steele et al. present a system for mobile devices which dynamically adapt presentations depending on a set of context properties that can be checked through the SMIL language.

A general framework for adapting XML documents to the device and to the user preferences is presented in [29, 30], based on the separation of the document content from the presentation with a suitable DTD, and on the transformations supported by an XSLT processor.

In [17, 18] a device independent model based on the Universal Profiling Schema (UPS) is defined in order to achieve automatic adaptation of the content based on its semantic and on the capabilities of the target device.

The reviewed works, while exploring a wide range of context features, do not approach in a unified framework all the facets of multimodal document adaptation. Most of them analyze only a part of the context, generally the part related to equipment and resources, and do not provide a comprehensive, uniform framework for adapting the document content, structure and presentation to the variants of user, equipment and environment properties. Moreover, context analysis is in most cases limited to the handling of independent features, while in real situations different context facets should be considered in their mutual relations. For example, audio contents might be not appropriate for very different reasons: because the environment is noisy, like a crowded street, or because it requires silence, like a library. They might also be not appropriate because the equipment does not support them, or because the user is hearing impaired. The same kind of adaptation could therefore be suggested by very different, even opposite, contexts.

The same problems arises with user profile, i.e., preferences, language and physical abilities. In general context interpretation should be a complex, logic-based process generating constraints to which to bind the multimodal document content and structure.

Our point of view is in fact more general, and is closer to the approach of Human Computer Interaction, where the ultimate goal of context-awareness is to allow a user to receive information with content and presentation suitable for the situation in which he/she is, spending an adequate interaction effort [31].

3. Document design

Adaptable document design concerns both the document structure and the adaptation parameters. In the multimodal domain three different descriptions must be provided:

- the static architecture, defining the structure of the document's content;
- the dynamic architecture, describing the media behavior along time, their mutual synchronization relationships and the reactions to user interaction;
- the context dependencies, linking the document components to the context dimensions, thus allowing the document management system to deliver the proper combination of media, content and presentation in any context.

Several document models provide the foundations for the above issues. We ground our presentation on a multimedia document model we have defined in a previous work [13] and experimented in several application scenarios, which is here briefly surveyed and extended to consider adaptation. Only the basic properties of the model will be presented, so the discussion applies also to other multimodal document models with minor changes.

3.1. Static architecture

A multimodal document is a collection of media items, which can be *atomic* or *composite*. An *atomic* item is a self-contained chunk of information, materialized in a media file. *Continuous media* (audio and video) have a temporal behavior, therefore they can act as triggers in the dynamic architecture. *Static media* (text and images) can be constrained to follow the dynamic behavior of continuous media. A *composite* item is a collection of items which has observable dynamic properties as a whole. Composite items can contain other composite items in a hierarchical schema. The document itself is a composite item, and can be used as a building block to design more complex documents.

Since an adaptable document can be delivered in different contexts and can be played with different modalities on different devices, generally we need to define more component items than effectively used in each context.

In adaptable documents *virtual* components are defined, which can be instantiated into *concrete* components (atomic and composite items) according to the context, as described in Section 5.

3.2. Dynamic architecture

Temporal constraints define the dynamic behavior of the multimodal document. The model we have defined describes the synchronization relationships among the document components in an event-based fashion.

A small number of synchronization relations define media object reactions to starting and ending of continuous media items which rule the presentation timing, and to user actions on media. In order to settle a simple background for the discussion of context-awareness, we limit our discussion to the basic behavior of the two relations describing parallel and sequential execution of media items. The forthcoming discussion therefore can be applied as well to documents described by other models

like, e.g., SMIL, even if some differences exist [13] which are not relevant in the context of this paper.

The parallel composition of media items a and b is described by the relation “ a plays with b ,” symbolically written $a \Leftrightarrow b$: if a or b is activated by the user or by some event, the two items play together. Item a acts as a “master” in this relation: its termination forces the termination of b .

The sequential composition of media items a and b is described by the relation “ a activates b ,” symbolically written $a \Rightarrow b$: when object a ends, object b begins playing.

Since static media have no timing properties, they cannot appear in the left side of the above relations, and their actual duration is ruled by synchronizing them with continuous media. Two special items, the *timer* and the *button*, are defined to give them a dynamic behavior independent from other media: a timer is a continuous media item with a predefined time length, while a button is a continuous media item which ends when the user activates it, e.g., with a mouse click.

Finally, composite items can be source and destination of synchronization relationships; in particular, a composite item ends when all the enclosed media end.

Figure 1 summarizes the symbology used for representing the components of a multimodal documents.







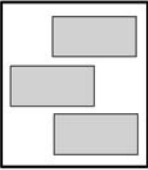
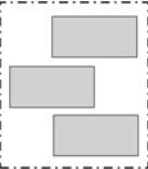

	<p>An atomic item; the icon describes the media type:  video,  audio,  text and  image.</p>
	<p>A button for user control of media timing.</p>
	<p>Composite media items are represented by thick rectangles enclosing the component items.</p>
	<p>Virtual components are represented by dashed rectangles enclosing the context dependent items.</p>
	<p>Synchronization relationships label the arrows connecting the involved media items.</p>

Fig. 1 The graphical symbols used to represent adaptable multimodal documents

3.3. Context constraints

Context constraints describe how media items are selected for building an instance of a multimodal document. A set of features is associated to each component, which describes the media requirements for delivery and presentation. Similarly, the context is described by a set of properties concerning the device, the network, the user and the environment. The compatibility between the context properties and the media features is defined by a rule system, whose evaluation makes suitable media to be identified for adapting the document to the context.

A virtual components is compatible with the union of contexts to which media items it contains are compatible; a composite is compatible with the contexts which correspond to the intersection of compatible contexts for media items it contains. A context-aware multimodal document can be instantiated in a given context if all its components are compatible with that context, in a recursive process.

Due to context variants, an item can be defined as mandatory or optional, and can be context independent, context dependent or context selectable; these terms will be discussed in Section 5.

4. A case study

As a case study we analyze a document presenting a meteorological forecast. Even if it is a very simple document, it can be adapted to several situations with non-trivial variants in content, dynamics and presentation.

The document is divided in two parts, a description of the current meteorological situation, and the forecast for the following hours. The two parts are delivered and presented to the user in sequence.

We consider five situations involving different environments and devices:

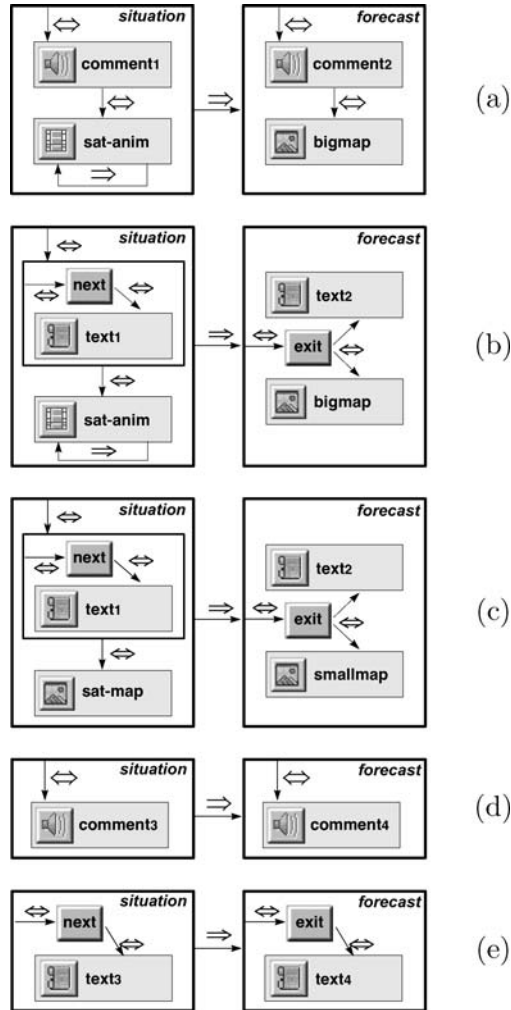
- (a) a desktop or notebook computer in a home or office environment;
- (b) a computer as above, but without audio capability, e.g., because the environment requires silence and no personal audio device is available¹;
- (c) a PDA provided with a wireless connection with limited bandwidth, used in an environment requiring silence;
- (d) a cellular phone able to receive only audio and short text messages, receiving an audio message;
- (e) the same cellular phone receiving a series of SMSs.

Figure 2 illustrates five different versions of the meteorological forecast document, adapted to the situations (a)–(e) above. Each version is adapted in content (short/long descriptions, large/small images), media type (video animations, images, audio, text), user device (a desktop computer, a PDA, a cellular phone), and situation (silence). The cases we discuss and the related documents are therefore significantly different.

Other adaptations concerning the presentation language and the physical media formats could contribute to tuning the document content and format to the user needs, to the communication infrastructure and to the device capabilities, but the

¹ Some situations are tailored to the discussion of adaptation and context-awareness issues within the space of this paper, therefore they may appear somehow artificial with respect to real cases.

Fig. 2 Different versions of a meteorological forecast document



differences are less relevant and do not affect its conceptual design. In Section 8 we shall introduce some issues about this further level of adaptation.

In the five situations above, five different documents should be issued:

- (a) A multimedia document with full audio and animated video. An animated satellite view loops as long as an audio describes the current situation; when the audio ends, a large forecast map is displayed and described by another audio comment. The synchronization is based on the time length of the audio. At the end of the forecast audio comment the whole document ends.
- (b) In lack of audio capabilities, comments must be displayed as texts. The document is therefore suitable for silent environments. In this case the synchronization structure of the document cannot be based on the intrinsic timing properties of media: in fact, the text has no defined time length (in our model it is infinite), and the length of the animated satellite view cannot be

used as a timer, because it is not related to the time needed for reading the text, which depends on the user pace. Therefore the synchronization must be based on an explicit user interaction, through a button pressed when the user has finished reading the text, which terminates the text display and the animation. The user manually advances to the second part where the text and the map describing the forecast are displayed; when the user presses an *exit* button, the whole presentation ends.

- (c) In the PDA version, due to the limited capabilities of the PDA and of the network, the animation is replaced by a small image; the forecast map of the second part of the document is also a small image, hence different from cases (a) and (b). As in case (b), there is no audio and the user controls the timing by advancing manually from the first part of the document to the second, where the structure of case (b) above is replicated.
- (d) An audio only document can be delivered to a cellular phone as a sequence of two messages, which do not require user intervention because they have their own timing. The splitting in two parts is functional to having the same overall structure for the whole document, which is important at design time. At delivery time the distinction for the user is less evident. The comments are plausibly different, in length and level of detail, from the ones delivered in case (a), due to the lack of accompanying images.
- (e) A text-only version is designed for being delivered to a cellular phone as a sequence of two short messages (SMS). The user steps through the messages with a button. The texts are plausibly different from the ones delivered in cases (b) and (c) due to size limitations of SMS messages.

It is easy to understand that the documents presented do not represent the only possible solutions. E.g., the synchronization schema of the *forecast* composite in case (b) could be the same as the *situation* composite; their behavior as perceived by the user would be the same, and different design strategies could naturally lead to either choice.

It is also obvious that simpler documents could be issued in rich contexts too. Cases (b)–(e) are compatible with the context of case (a); they simply do not use all the features of the device and of the environment, giving a more limited, but anyway correct, information to the user. The choice of the “correct” (or, better, “most suitable”) version is therefore the result of an evaluation process which maximizes some function of the information carried by the media against the constraints expressed by the context.

The five variants described above can be designed as a unique compound document (called *virtual document*) whose components are selectively instantiated under different contexts, leading to adapted documents which differ in content and presentation, and may differ in structure according to the mutual relationships among the instantiated components. Each component is tagged by features that can be related to the context in which the document is delivered. In figure 3 a pictorial symbolic representation of a virtual meteorological forecast document is shown. Letters in grey circles identify which are the relevant cases of figure 2 to which each media item applies; dashed rectangles enclose alternative components, that are selected according to the context for instantiating a specific adapted document. The structure of the *situation* composite is the sum of the structures defined in cases (a)–(e); in the *forecast* composite a uniform control structure is defined, which associates

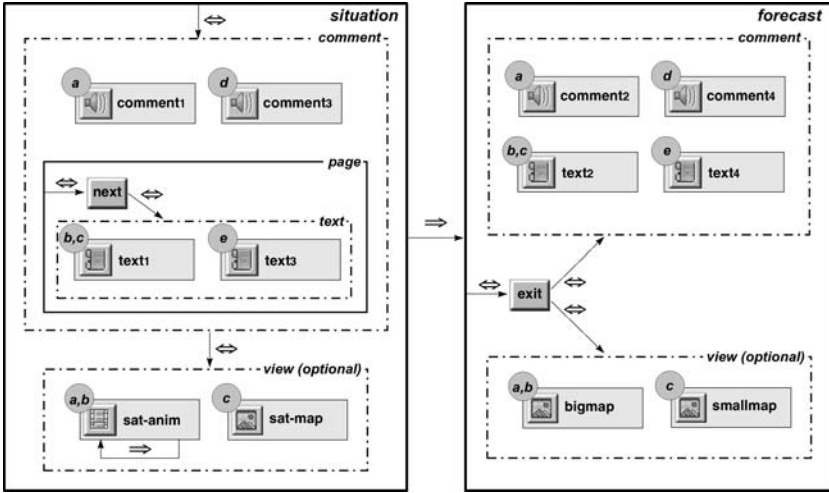


Fig. 3 An adaptable meteorological forecast document

the *exit* button to the end of the forecast in all the cases. The meaning of the virtual document components and how they are related with the context will be described in the following two sections.

Like the design of a concrete document, also the design of a virtual document is subject to variants. For example, in figure 4 a variant of the virtual document of figure 3 is shown: in the *situation* composite, the different alternatives of the *comment* part, both for text and audio, are exposed at the same level, while in figure 3 the differences in the synchronization structure are represented at a level higher than the differences between the text page contents. The structure of the *forecast* composite replicates the structure of the *situation* part, while in figure 3 a more uniform control structure was designed. We do not approach here the problem of making a “good” design of adaptable documents, an important issue that should be

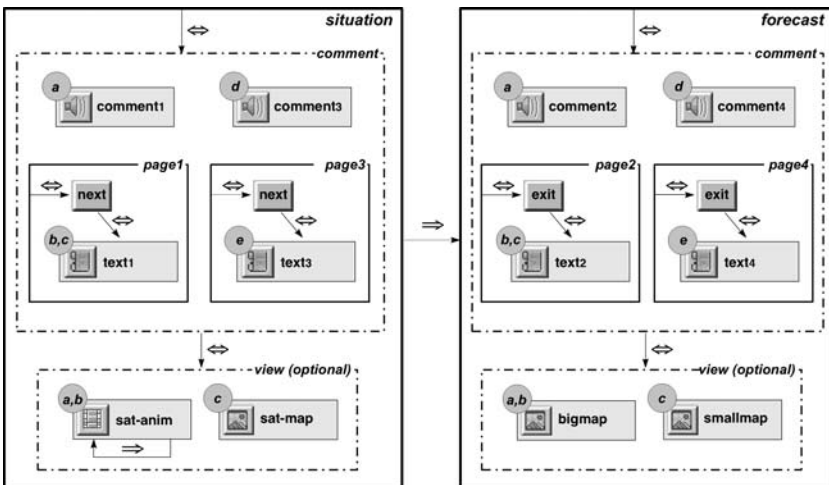


Fig. 4 A variant of the document of figure 3

supported by suitable guidelines and methodologies. Different virtual component structures and organizations should be considered equivalent with respect to the instantiation process if concrete documents generated by different virtual documents have the same content and dynamics, i.e., they are behaviorally equivalent as defined in [3].

5. Adaptation design

An adaptable multimodal document is a *virtual document*, made of *virtual components*, according to a structure which is independent from the context at macroscopic level, but may depend on it in its details. Each virtual component is a collection of composite and atomic media items whose features allow the adaptation system to select the proper ones as a function of the context. The instantiation of a virtual document into a *concrete document* consists in the identification of the document virtual components and, for each of them, in the selection of the proper *instances* compatible with the given context.

In order to make a translation from a virtual to a concrete document, three descriptions must be given:

- how to characterize the different variants, i.e., how to describe the features of the component media items;
- how to characterize the context;
- how to identify, for each context, the correct or relevant variant.

Virtual documents are described as XML documents conforming to a schema which defines both the static and the dynamic structure of the documents. The XML schema for context-aware multimodal documents extends the one we have defined for multimedia documents. The XML document is made of three sections: the `<layout>` section contains the definition of the *channels* used by the media objects, e.g., the position and size on the screen of the visual components, the `<components>` section describes the media and the static structure of the document, and the `<relationships>` section contains the temporal constraints between the media items. Media features are not described in the XML document. They are associated to media in a media repository, as described in Section 6. The most relevant section for the focus of this paper is the *components* section that describes the virtual components of the document and their instances.

A virtual component is defined by the tag `<vcomponent>` and encloses a list of items with different features, which apply to different contexts. Figure 5 shows a fragment of the XML document describing the virtual document of figure 3, where non-relevant details are omitted. The pictorial representation of figure 3, in fact mirrors the XML description²: virtual components are denoted by dashed frames which enclose the composite and atomic media subject to context dependence. From the synchronization point of view, a virtual component behaves as a *black box*: synchronization relationships can be established only between the virtual component and other components external to it; they will be inherited by the

² The names of the media items, composites and virtual components must be unique, therefore a hierarchical naming scheme is adopted in XML; in figure 3 a simple naming schema is used, based on the visual interpretation of the components hierarchy.

```

<?xml version="1.0" encoding="UTF-8"?>
<presentation xmlns="adapt_presentation.xsd" title="meteo">
...
  <components>
    <module id="meteo">
      <composite id="situation">
        <vcomponent id="situation_comment">
          <composite id="situation_comment_page">
            <ui id="next" />
            <vcomponent id="situation_comment_page_text">
              <page type="text" id="text1" file="..." ... />
              <page type="text" id="text3" file="..." ... />
            </vcomponent>
          </composite>
          <clip type="audio" id="comment1" file="..." ... />
          <clip type="audio" id="comment3" file="..." ... />
        </vcomponent>
        <vcomponent id="situation_view" optional="yes">
          <clip id="sat_anim" type="video" file="..." ... />
          <page type="image" id="sat_map" file="..." ... />
        </vcomponent>
      </composite>
      <composite id="forecast">
        <ui id="exit" />
        <vcomponent id="forecast_comment">
          <clip type="audio" id="comment2" file="..." ... />
          <clip type="audio" id="comment4" file="..." ... />
          <page type="text" id="text2" file="..." ... />
          <page type="text" id="text4" file="..." ... />
        </vcomponent>
        <vcomponent id="forecast_view">
          <page type="image" id="bigmap" file="..." ... />
          <page type="image" id="smallmap" file="..." ... />
        </vcomponent>
      </composite>
    </module>
  </components>
...
</presentation>

```

Fig. 5 A fragment of the XML virtual document

concrete instance during the adaptation process. As a consequence, if relations such as $vc \Leftrightarrow i$ or $vc \Rightarrow i$ exist between a virtual component vc and a generic item i , all the items in the virtual component must be continuous.

A virtual component can be *mandatory* or *optional*, according to its role in the document semantics. This property is set by the attribute `optional` which holds the values `yes` or `no`. A mandatory virtual component carries a content which is necessary to understand the document. Therefore, at least one of its internal items must be instantiated, unless the virtual components itself is contained, directly or indirectly, into another virtual component with alternatives for different contexts. If the virtual components at the outermost level of the document structure cannot be instantiated, the concrete document is *incomplete* and cannot be delivered. An optional virtual component can be instantiated or not instantiated, according to its compatibility with the context. Its absence does not prevent the document from being understandable.

In figure 3, the virtual component *comment* is mandatory, while the virtual component *view* is optional, because it can be suppressed without loss of document significance. In fact, the information represented by it is not instantiated in cases (d) and (e).

The virtual component *text* in the composite *page* is also mandatory; if it cannot be instantiated because of context conflicts, the whole composite *page* cannot be instantiated. This does not prevent a concrete document from being generated, since alternatives are defined through the audio comments. However, if none of the media contained in the *comment* virtual component is compatible with the context, the instantiation process fails.

Generally speaking, media items are *context dependent*, since they exist in several instances for the different contexts, selected at document instantiation time. Some media, however, could be *context independent*, i.e., they could appear in any instance of a virtual document, either because their properties are compatible with any context (e.g., a short constant text message), or because they are strictly bound to other components; e.g., buttons and timers inherit the context dependencies from the media items they control.

Finally, in some cases several media items, instances of a same virtual component, could be compatible with the same context. We call them *context selectable* items. For example, texts at several degrees of detail and images with several resolution and size values are in principle compatible with contexts whose equipment has large display capabilities. In this case the user could be asked (but this solution is not optimal from the interaction point of view), or some intelligence could be put in the system in order to make a “good” choice. A wide range of approaches, from simple priority schemas up to intelligent agents, could give support to this issue, but we do not discuss them here for space reasons.

6. Context-awareness design

Figure 6 shows the proposed architecture of a context-aware document adaptation system, which is implemented in a demonstration prototype written in Java and SICStus Prolog [26]. The prototype implements a visual interface which allows the document designer to simulate different contexts and evaluate the corresponding concrete document. The resolver engine, as discussed in the following, is implemented in Prolog, using the Jasper Java interface to load the SICStus runtime kernel from the Java program which executes the virtual document analysis and the concrete document instantiation. The XML description of the virtual and concrete document is processed through the JAXB library (Java API for XML processing) [25].

A *media repository* holds the instances of all the media items and the description of their features.

The *context* is represented by a set of context properties. Several formalisms can be used for describing media and context properties, which are in principle equivalent as long as they can describe in a correct and complete way the relevant features and their values. The W3C CC/PP proposal [15] is a standard which allows designers to describe in an extensible way the context and media features in a simple structured attribute/value fashion. Other formalism may be used as well, e.g., based on MPEG 7 or on logic assertions. For example, Ranganathan et al. [20] propose a

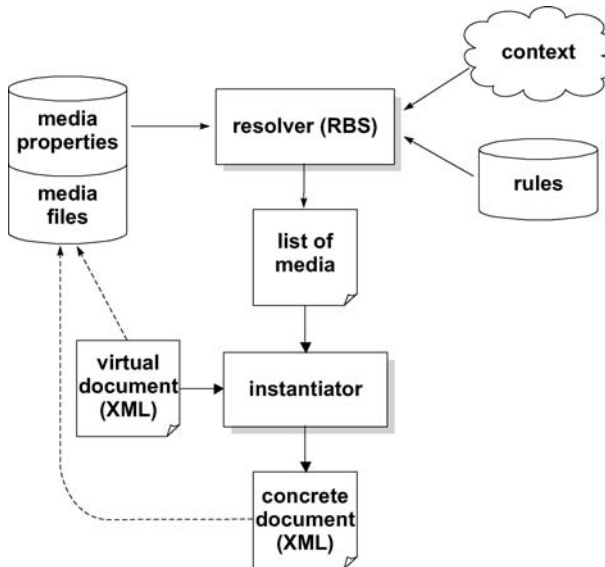


Fig. 6 The architecture of a context-aware document adaptation system

context model based on first order logic predicate calculus. We do not elaborate on this issue here, noting that the different formalisms should be considered semantically equivalent, and that translations are possible among them.

The association between the context properties and the media features is managed by a *resolver*, a rule-based system (RBS) which defines how media can be used in different contexts: given the set of features of the media contained in a virtual document and the set of context properties, it is able to select the media items that are compatible with that context.

The list of selected media items is used by the system to instantiate a *concrete document*, which contains only the compatible media, correctly structured. The *layout* and *relationships* sections of the XML virtual document are processed according to the media items that survive the adaptation process. Without entering into too much detail, it is sufficient to say that in the *layout* section only the channels used by the selected media are retained, while in the *relationships* section only the temporal relations connecting components that have been instantiated are retained. An example will be discussed later in this section.

The resolver is implemented with a Prolog engine, therefore media features and context properties should be translated into Prolog predicates if they are expressed in other formalisms. In the current version of the prototype, since we are focused on the adaptation process rather than in media description, context properties and media features are expressed as logical assertions. Figure 7 shows a sample of media features, device properties and adaptation rules referred to the example document so far illustrated, with minor details omitted and replaced by ellipses (...).

The generation of a document instance compatible with a given context is made in several steps which examine the different context facets, process the mandatory and optional components, possibly select among different alternatives the “optimal” one, and build the concrete document structure and its synchronization relationships.

```

Media Features
type(sat-anim, video)                format(sat-anim, mpeg1)
type(comment3, audio)                 format(comment3, mp3)
type(text1, text)                     format(text1, ascii)
type(smallmap,image)                 format(smallmap, gif)
screen_size(smallmap, 160, 120)

Device Features
audio(desktop)                        audio(pda)
scr_size(desktop,1024, 768)            scr_size(pda,320, 240)
supported_format(desktop,[. . .,mpeg1,gif,ascii,mp3,. . .])
supported_format(pda,[avi,gif,bmp,ascii,mp3,wav])

Adaptation rules
adapt_video(D, U, M):-
    . . .
    adapt_screen(D, M), type(M,image),
    screen_size(M,SX,SY), scr_size(D,DX,DY),
    SX ≤ DX, SY ≤ DY.
adapt_screen(D, M) :-
    graphic_video(D,yes),
    format(M,F), supported_format(D,SF), member(F,SF),
    . . .
adapt_env(D, A, M):-
    (sound(A,silence) ; sound(A,noisy)), (headset(D,yes),
    type(M,audio)).

```

Fig. 7 A sample of media features, device properties and adaptation rules

1. Pick up the media items which are compatible with the user device; e.g., if the device has not audio capabilities, the concrete document cannot contain audio files. In figure 7 the function *adapt_video* shows a sample of the rules that check if the size and the format of an image (M) is compatible with the screen size of the device (D) and, through the function *adapt_screen*, if the device has graphic capabilities and the media format (F) is among the supported ones (SF).
2. Process information about environment. This step can be considered a refinement of the selection made on the device: e.g., if the user is in a room which requires silence, or in a very noisy environment, audio files cannot be delivered unless the device is provided with headsets (function *adapt_env* of figure 7).
3. Select from the list of the retrieved items those compatible with the user profile, such as the language, the abilities (e.g., a sight impaired user can receive only audio messages) and the preferences (e.g., level of detail).

At the end of this step the list of media compatible with the context is built. The following steps examine the document static architecture in order to instantiate a correct and complete concrete document.

4. For each mandatory virtual component, check that it can be instantiated, i.e., at least one instance of it exists for the specified context; otherwise no concrete document can be generated for that context.

This process is executed recursively on the component hierarchy according to these rules: (a) a concrete component can be instantiated; (b) a composite can be instantiated if all its constituent components can be instantiated; (c) a virtual component can be instantiated if at least one of its alternatives can be instantiated. If more than one instance exists, the component can be instantiated but additional processing is required to select the proper instance, as explained below.

5. For each optional virtual components check if at least one concrete instance exists for the specified context. If it exists, the component can be instantiated, otherwise it is removed from the document.
6. If context selectable instances exist, make a further selection using some priority mechanism, intelligent assistant, heuristic evaluation, or by asking the user.³
7. As long as virtual components are instantiated, synchronization relationship are transferred from them to the concrete instances; in detail, for each relation $A \Leftrightarrow b$ or $A \Rightarrow b$, where A is a virtual component, if a is the instance generated by A , the relation is replaced by $a \Leftrightarrow b$ or $a \Rightarrow b$, respectively. The same transformation applies to relations where the second element or both the elements are virtual components.

As an example of adaptation, we consider the meteorological forecast document introduced in figure 2, adapted to a PDA with a small screen size and no animated video capabilities (situation (c) of figure 2). We assume the user is in a library, i.e., a silent environment, and does not wear headsets, therefore cannot receive audio information even if the device supports it.

The instantiation proceeds according to the following steps:

1. Pick up the media compatible with the device. Considering the structure depicted in figure 3, the system selects the media items compatible with the PDA device: the text files, the satellite image (*sat-map*) and the small forecast image (*smallmap*). The animation (*sat-anim*) is discarded because the PDA does not support it, and the large forecast image (*bigmap*) is discarded because it does not fit the size of the PDA screen.
2. Consider information about the environment. Since the library requires silence and the user has no headsets, the system discards the four audio comments.
3. Select the items compatible with user preferences. In our example the context is not differentiated on user preferences, therefore no media item is discarded. In a real life case, language, level of detail, narration style, etc., could be analyzed for selecting text items, while user abilities like the ability to perceive correctly the colors, could affect the selection of images.

The virtual document structure at this stage of processing is pictorially described in figure 8. It contains the four texts, the satellite image (*sat-map*) and the small forecast image (*smallmap*). The processing of the structure and the instantiation of the concrete document follow.

4. The mandatory components are recursively examined; we unfold the recursion in a bottom-up style for ease of description: (a) the mandatory virtual component *text* can be instantiated, since it contains only concrete media instances; (b)

³ The details are not relevant since the instances are assumed logically equivalent for that context.

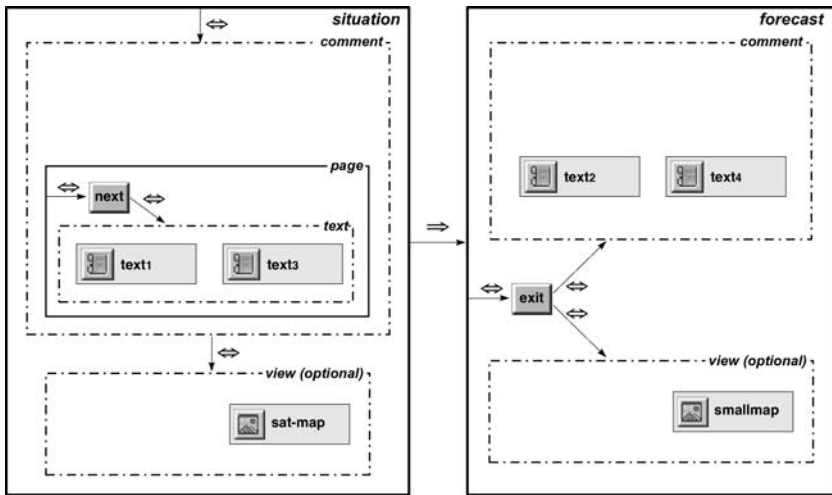


Fig. 8 Context compatible media instances in the virtual meteorological forecast document

therefore, the composite *page* can be instantiated; (c) therefore, the mandatory virtual component *comment* can be instantiated in the selected context.

In the *forecast* element, the mandatory virtual component *comment* can be instantiated because it contains only concrete media instances.

5. The optional components are examined: In the *situation* element, the virtual component *view* can be instantiated because it contains the concrete media instance *sat-map*; similarly, in the *forecast* part the virtual component *view* can be also instantiated.
6. The document obtained so far contains two pairs of two instances of *context selectable* items, i.e., the text files, which must be uniquely selected. A good choice, driven by an evaluation process which is not described here, would pick *text1* and *text2*, which describe the meteorological situation at a deeper level of detail, and can fit the PDA display.
7. Once the system has retrieved all the compatible media items, the instantiator builds the concrete document by instantiating the dynamic structure of the presentation, i.e., the synchronization relations. The synchronization relations which involve instantiated virtual components are retained, and are passed from the virtual component to the instance; the relations which involve virtual components which have not been instantiated are removed; the relations involving only composites and atomic media (e.g., the relation *situation* ⇒ *forecast*) are obviously not modified during the instantiation process. Table 1

Table 1 Instantiation of virtual synchronization relations

Virtual relation	Concrete relation
Situation ⇔ comment	Situation ⇔ page
Comment ⇔ view	Page ⇔ view
Next ⇔ text	Next ⇔ text1
Exit ⇔ comment	Exit ⇔ text2
Exit ⇔ view	Exit ⇔ smallmap

```

<?xml version="1.0" encoding="UTF-8"?>
<presentation xmlns="adapt_presentation.xsd" title="meteo">
  ...
  <components>
    <module id="meteo">
      <composite id="situation_comment">
        <composite id="situation_comment_page">
          <ui id="next"/>
          <page type="text" id="text1" file="..." ... />
        </composite>
        <page type="image" id="sat_map" file="..." ... />
      </composite>
      <composite id="forecast">
        <ui id="exit"/>
        <page type="text" id="text2" file="..." ... />
        <page type="image" id="smallmap" file="..." ... />
      </composite>
    </module>
  </components>
  ...
</presentation>

```

Fig. 9 A fragment of the concrete XML document

illustrates the changes made. The final document instance, which corresponds to case (c) of figure 2, is shown in figure 9.

7. Discussion

The first three steps of the instantiation process described in Section 6 select the concrete instances of media items on the basis of the features of the device of the user (step 1), of the user preferences (step 2) and of the information about ambience (step 3). The process does not impose any priority upon this information, therefore the order in which they are considered can be changed without affecting the final result.

However, it may be the case that a concrete instances of the document does not exist for a particular context, e.g., only audio files can be used, but the document contains audio items whose format is not supported by the device, or text items are not available in the language known by the user. In this case, a set of *relaxing* rules can be defined to impose priority over the constraints defined in the resolver, thus introducing a sort of *transgressive* context adaptation [10] which allows for the delivery of the document in more contexts. E.g., constraints about device's capabilities must be considered, but information about user preferences and ambience conditions can be ignored, or can allow some tolerances, in order to retrieve a document.

Additional processing could be done during and after the instantiation of a concrete document, which is not illustrated in figure 6, to optimize the adaptation to mobile environments, and to environments with limited resources capabilities. E.g., building the concrete document structure is a simple task, keeping in the virtual document structure only the list of concrete media obtained by the resolver. However, a complex document could be delivered as a whole in contexts with abundant resources and a fixed environment, while it could require a progressive delivery of sequential fragments in limited or varying resource contexts, or where the user situation can change during a session.

As an example, a user moving in a museum with a portable device could ask for a detailed description of an artwork, made of text, audio and images compatible with the device, which are better delivered in chunks; for each chunk the resources availability is checked, and the user is asked to confirm for receiving further information, in order to avoid to be stuck in long downloads. Delivering a document could therefore require to split it into parts (modules, sections, etc.) which are instantiated and delivered separately. Each part is identified by a *main component* which is the item which starts the playback of that part. Some issues relevant to this case are discussed in a previous work by the same authors [6].

8. Conclusion

We have presented a context-aware framework for designing multimodal documents adaptable to different user and resource situations. Context-awareness results in the definition (at design time) and selection (at delivery time) of a set of concrete document components according to a multimodal document model which associates components to context information through a rule system.

Among the issues that deserve further investigation, the consistency of a complex document under different context conditions is of primary importance. For example, in two different virtual components, each containing several context-selectable instances, only specific combination of instances could be meaningful for the user, hence limiting the number of cases to be examined.

Consistency can be approached by additional rules that describe mutual consistency relationships, as much as in a traditional database referential integrity defines mutual constraints between database records.

The adaptation model described in this paper could be extended to a two phase model in order to consider more flexible adaptation strategies and processes: a first phase of *static* adaptation, which is the one described in this paper, could be followed by a *dynamic* adaptation phase in which the content and the presentation of the components of the concrete document are adapted with a fine grain view to the context, mainly to the device properties and to the available system resources. The type of the components of the concrete document is not changed during the dynamic adaptation, nor is the structural and synchronization relationships among them. The content and the presentation can, however, be processed only at this stage, starting from prototypical media instances, that can be abstracted, compressed, resized, translated, etc., without changing their type, meaning and role in the document architecture. In such a way, a wider range of context values and facets can be considered without designing all the possible media instances in advance.

The rules associating the media to the context should be split in two sets: besides the rules defining the association between media features and context properties as described here, which should not consider low-level physical properties like image size, coding format or text length, a second set of rules must define how and at what extent media content and presentation can be transformed by functions, programs or services that can be invoked at delivery time. Such functions, programs and services can execute a wide range of operations from image resizing to text translation and abstraction, and are already used in many of the systems reviewed in Section 2.

The instantiation process performs the selection of the media instance prototypes according to the possibility of adapting them to their final version, also evaluating the cost of such adaptation if multiple choices exist (e.g., choosing between the abstraction of a text in the correct language, or the translation of an already abstracted text).

Acknowledgment We gratefully acknowledge Sabina Rossi for her support in the design of the rule base system.

References

1. Abowd GD, Atkeson CG, Hong J, Long S, Kooper R, Pinkerton M (1997) Cyberguide: a mobile context-aware tour guide. *ACM Wirel Network* 3:421–433
2. Bertino E, Ferrari E, Stolf M (2000, January) MPGS: an interactive tool for the specification and generation of multimedia presentations. *IEEE Trans on Knowl and Data Eng* 12(1):102–125
3. Bertolotti P, Gaggi O (2004) A notion of equivalence for multimedia document. In: *Proc. Int. Workshop on Multimedia Information Systems*, Washington, District of Columbia, pp. 96–104
4. Boll S, Klas W, Wandel J (1999) A cross-media adaptation strategy for multimedia presentations. In: *ACM Multimedia*, pp. 37–46
5. Brusilovsky P (1996) Methods and techniques of adaptive hypermedia. *User Model User-Adapt Interact* 6(2–3):87–129
6. Celentano A, Gaggi O, Sapino ML (2004) Retrieval in multimedia presentations. *Multimedia Syst* 10(1):72–82
7. Chen G, Kotz D (2000) A survey of context-aware mobile computing. Technical Report TR2000–381, Dartmouth College, Department of Computer Science
8. Davies N, Cheverst K, Mitchell K, Efrat A (2001) Using and determining location in a context-sensitive tour guide. *IEEE Comput* 34(8):35–41
9. Dey AK (2001) Understanding and using context. *Personal Ubiquitous Computing* 5(1):4–7
10. Dias V, Euzenat J, Layaïda N (2003) A semantic framework for multimedia document adaptation. In: *Proc. 18th Int. Joint Conf. on Artificial Intelligence*, pp. 31–36
11. Elouazizi N, Bachvarova Y (2004) On cognitive relevance in automatic multimodal systems. In: *Proc. IEEE Conf. on Multimedia Software Engineering*, Miami, Florida, pp. 418–426
12. Fink J, Kobsa A, Nill A (1998) Adaptable and adaptive information provision for all users, including disabled and elderly people. *New Rev Hypermedia Multimed* 4:163–188
13. Gaggi O, Celentano A (2002) Modelling synchronized hypermedia presentations. *Multimedia Tools Appl* 27(1):53–78
14. Harter A, Hopper A, Steggles P, Ward A, Webster P (1999, August) The anatomy of a context-aware application. In: *Proc. 5th Annual ACM/IEEE Int'l Conf. on Mobile Computing and Networking*, Seattle, Washington, pp. 59–68
15. Klyne G, Reynolds F, Woodrow C, Ohto H, Hjelm J, Butler MH, Tran L (2004) Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies 1.0. W3C Recommendation
16. Kraus M, Bry F, Kitagawa K (2003, December) XML document adaptation queries (XDAQ): an approach to adaptation reasoning using Web query languages. In: *Proc. Workshop on Principles and Practice of Semantic Web Reasoning (PPSWR)*
17. Lemlouma T, Layaïda N (2003) Adapted content delivery for different contexts. In: *IEEE Symposium on Applications and the Internet (SAINT 2003)*, Orlando, Florida, pp. 190–199
18. Lemlouma T, Layaïda N (2004) Context-aware adaptation for mobile devices. In: *Proc. IEEE Int. Conf. on Mobile Data Management*, Berkeley, California, pp. 106–111
19. Petrelli D, Not E, Zancanaro M, Strapparava C, Stock O (2001) Modelling and adapting to context. *Personal Ubiquitous Computing* 5(1):20–24
20. Ranganathan A, Campbell RH (2003) An infrastructure for context-awareness based on first order logic. *Personal Ubiquitous Computing* 7(6):353–364
21. Schilit BN, Adams N, Want R (1994, December) Context-aware computing applications. In: *Proc. Workshop on Mobile Computing Systems and Applications*. IEEE

22. Schmidt A, Beigl M, Gellersen Hans-W (1999) There is more to context than location. *Comput Graph* 23
23. Schneider-Hufschmidt M, Kühme T, Malinowski U (eds) (1993) *Adaptive user interfaces: principles and practice*. Elsevier Science, Amsterdam
24. Steele R, Lubonski M, Ventsov Y, Lawrence E (2004) Accessing SMIL-based dynamically adaptable multimedia presentations from mobile devices. In: *International Conference on Information Technology: Coding and Computing (ITCC'04)*, Volume 2, Las Vegas, Nevada, USA, pp. 410–415
25. Sun Microsystems. *Java Architecture for XML Binding (JAXB)*. <http://java.sun.com/xml/jaxb/>
26. Swedish Institute of Computer Science. *SICSStus Prolog*. <http://www.sics.se>
27. Synchronized Multimedia Working Group of W3C (2001, August) *Synchronized Multimedia Integration Language (SMIL) 2.0 Specification*
28. van Ossenbruggen J, Geurts J, Cornelissen F, Rutledge L, Hardman L (2001) Towards second and third generation web-based multimedia. In: *Proc. Tenth International World Wide Web Conference*, Hong Kong, pp. 479–488
29. Villard L (2001) Authoring transformations by direct manipulation for adaptable multimedia presentations. In: *Proc. 2001 ACM Symposium on Document Engineering*, Atlanta, Georgia, USA, 2001, ACM, pp 125–134
30. Villard L, Roisin C, Layaïda N (2000, September) A XML-based multimedia document processing model for content adaptation. In: *Proc. of Digital Documents and Electronic Publishing (DDEP00)*, vol. 2023 of LNCS, Munich, Germany. Springer, Berlin Heidelberg New York
31. Weiser M, Seely Brown J (1997) *The coming age of calm technology*. In: *Beyond calculation: the next fifty years of computing*. Springer, Berlin Heidelberg New York
32. Wirag S (1997) Modeling of adaptable multimedia documents. In: *Proc. Int. Work-shop on Interactive Distributed Multimedia Systems and Telecommunication Services (IDMS'97)*, Darmstadt, Germany, pp 220–230



Augusto Celentano is full professor of Information Systems at Ca' Foscari University in Venice. He received a Master Degree in Electronic Engineering from the Politecnico di Milano. He has been Head of the Department of Computer Science of Ca' Foscari University, and Deputy Rector for information technology. Augusto Celentano has been member of scientific committees in research and educational centers of the Politecnico di Milano, and has been consultant and scientific coordinator of research projects of the European Union. His current research interests are in mobile, pervasive and context-aware information systems, in multimedia systems and in Human Computer Interaction. He is co-author of more than 80 scientific papers published in international journals and conference proceedings, and is serving as reviewer and program committee member in international conferences.



Ombretta Gaggi received her PhD in Computer Science from the Consortium of Universities of Bologna, Padova and Venice in 2003, and is currently a Research Assistant at the Department of Pure and Applied Mathematics of the University of Padova. Her research interests include multimedia document and application design, mobile context-aware information systems, and distributed multimedia modelling. She is serving as reviewer and program committee member in international conferences.