Context-aware design of adaptable multimodal documents

Augusto Celentano and Ombretta Gaggi
Dipartimento di Informatica, Università Ca' Foscari di Venezia
Via Torino 155, 30172 Mestre (VE), Italia
{auce,gaggi}@dsi.unive.it

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.

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.

Adaptation is tightly related to context-awareness, and sometimes the two terms are used as synonyms. However, they denote two distinct concepts. In the document domain[1] *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 [5]: 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's 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 up to hypermedia navigation;
- a *multichannel* document can be delivered, in different formats and with different content versions, through different communication channels and different presentation devices, such as wide band or GPRS networks, for 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 fol-

1 Adaptation and context-awareness are also emerging features of applications and services.

lowing 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 adaptation and context-awareness. The term context is mainly associated to the concept of location, but is far richer, as underlined by several works [7, 14, 15]. 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"* [4]. Such information includes several facets, related to the user situation (location, time and mobility, profile), to the environment (indoor/outdoor, light/darkness, sound/silence), to the communication (wide/narrow band, plugged/wireless, continuous/discontinuous), to the device (screen size and resolution, audio fidelity, input devices, user attention), etc.

Adaptation has been studied in domains like mobile computing, Web applications, user interfaces and hypermedia [1, 3, 10, 16]. Proposals exist for automatic adaptation of multimedia documents, even if they are limited to some aspects and do not involve the whole process of document design and delivery. In most cases, adaptation is bound to device and resource analysis.

In [20] a temporal model for multimedia presentations is discussed, which adapt themselves to a context is described in terms of network bandwidth, device and available resources; the presentation behavior can vary within specified limits.

Euzenat et al. [8] 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 all the properties of the original document) are defined.

A general framework for adapting XML documents to the device and to the user preferences is presented in [17, 18], 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 [11, 12] 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.

All the cited works approach the problem of context-aware adaptation of complex documents along selected directions, which are analyzed with a narrow focus. Our point of view is more comprehensive and closer to the one of HCI, 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 [19].

3. Document design

Adaptable document design concerns both the document structure and the adaptation parameters. Three different descriptions must be provided:

- the static architecture, defining the structure of the document's content;
- the dynamic architecture, describing the media behavior in time and in response 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.

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 [6] and used in several application contexts, which is here briefly surveyed and extended to consider adaptation. The discussion, however, applies also to other multimedia 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, and can be used to build 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.

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

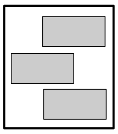
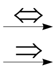
	An atomic item; the symbol describes the media type: video, audio, text, and image.
	A button for user control of media timing.
	Composite media items are represented by thick rectangles enclosing the component items.
	Synchronization relationships label the arrows connecting the involved media items.

Figure 1. The graphical symbols used to represent synchronized multimedia documents.

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 [6] 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, since 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 be sources in 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 contained media end.

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

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 concerning a meteorological forecast. Even if it is a very simple document, it can be adapted to several situations with non trivial variants in content, structure 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 normal 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 available[2];

(c) a PDA provided with a wireless connection with limited bandwidth;

(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 forcast document, adapted to the situations (a)–(e) above. Each version is adapted in content (short/long descriptions, large/small images), media (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.

2 Some situations are biased to the discussion of adaptiveness and context-awareness problems within the space of this paper, therefore they may appear someway artificial with respect to real cases.

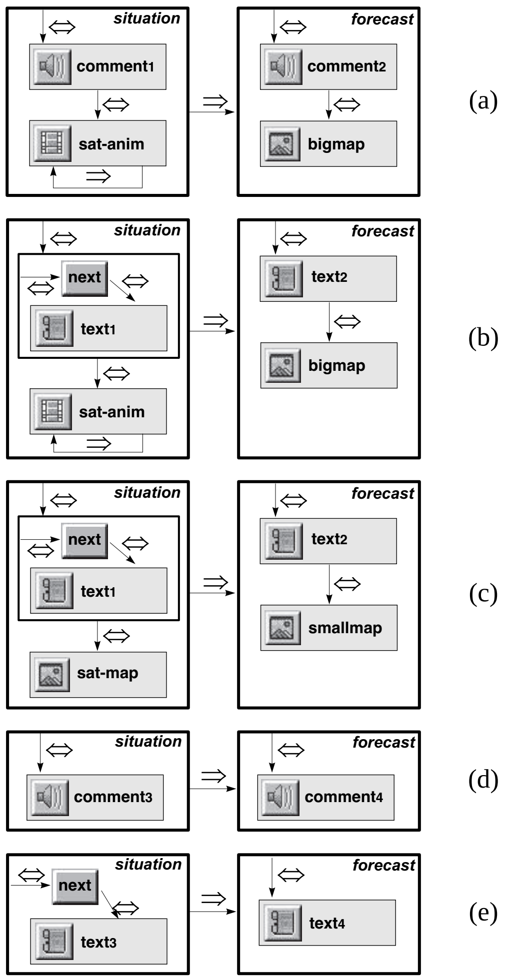


Figure 2. Different adaptations of a meteorological forecast document.

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 differences are less relevant.

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 audio ends, a 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 absence of audio capabilities, comments must displayed as texts. The document is therefore suitable for silent environments. In this case the synchronization structure of the document cannot be based on media timing properties: in fact, the text has no defined time length (in our model it is infinite), and the length of the animation cannot be used as a timer, because the user needs to read the whole message at his/her pace. Therefore the synchronization must be based on the user interaction, through a button pressed when the user has finished reading the text, which terminates the text display also. The user therefore manually advances to the second part, where the forecast is presented as in case (a).

(c) In the PDA version, due to the hypothetic 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), the user controls the timing by advancing manually from the first part of the document to the second.

(d) An audio only document can be delivered to a cellular phone as a sequence of two messages. 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 is less evident.

(e) This text-only version is designed for being delivered to a cellular phone as a sequence of two short messages (SMS). The user advances from the first message to the second with an explicit request.

5. Adaptation design

The five variants described in previous section 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 structure, content and presentation. 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 the meteorological forecast document is shown. Letters in grey circles identify which are the relevant cases of Figure 2 to which each media item applies. How they are specified is described in the following of this section and in the next one.

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 iden-

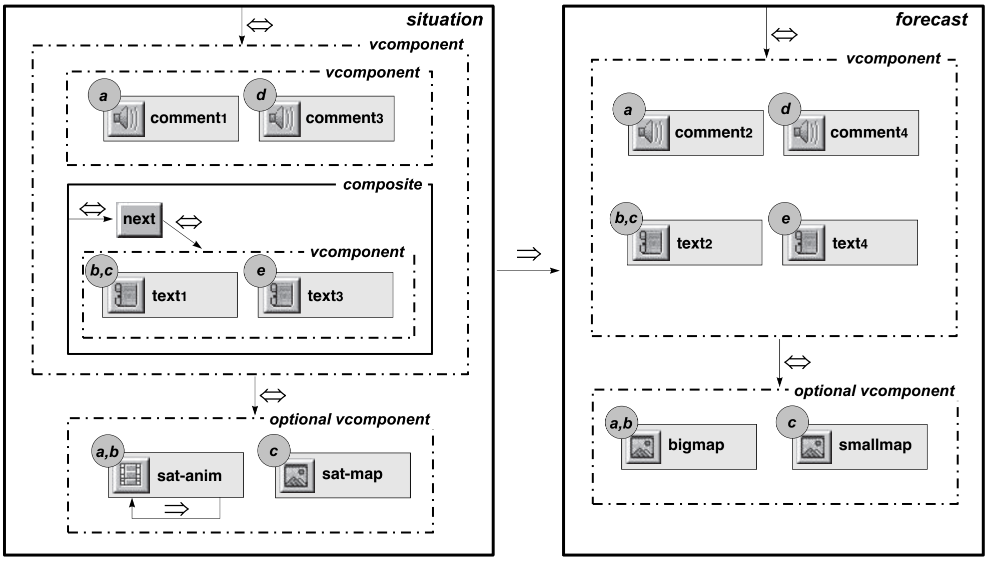


Figure 3. An adaptable meteorological forecast document.

tification 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 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. 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. In Figure 3 virtual components are denoted by dashed frames which enclose the composite and atomic media subject to context dependence. Synchronization relationships are established between the virtual components, and will be inherited by the concrete instances during the adaptation process.

A virtual component of an adaptable multimodal document can be mandatory or optional, according to its role in the document semantics. This property is expressed by the attribute optional. A mandatory component must always be present in the instantiated document for any context; if it is missing, the document is *incomplete* and cannot be delivered. An optional component can be instantiated and delivered or not, according to its compatibility with the specific context attributes during the concrete document instantiation. Its absence does not prevent the document from being understandable. E.g., in Figure 3 the upper virtual components are mandatory, while the lower ones are optional, because they are not instantiated in cases (d) and (e).

Media features are not described in the XML document. They are associated to media in a media repository, as described in Section 6.

Generally speaking, media items are *context-dependent*, since they exists 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, or because the virtual document specifies no alternatives.

Finally, in some cases several media items, instances of a same virtual media, could be compatible with the same context. We call them *context-selectable* items. E.g., 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 op-

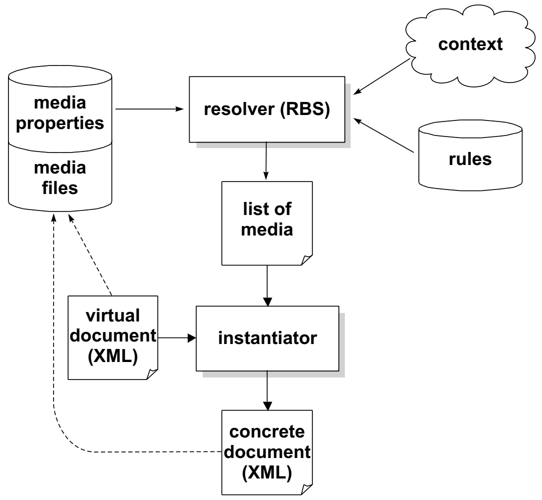


Figure 4. The architecture of a context-aware document adaptation system.

timal 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 4 shows the proposed architecture of a context-aware document adaptation system, which is implemented in a demonstration prototype written in Java and Prolog.

A *media repository* holds the instances of all the media items; as we have anticipated in previous section, media features are also contained in the repository.

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 [9] 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. [13] propose a context model based on first order logic predicate calculus. We do not elaborate on this issue here, noting that they 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 which connect components that have been instantiated are retained.

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 5 shows a sample of media features, device properties and adaptation rules referred to the example document so far illustrated, with minor details omitted (. . .).

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.

1. Pick up the media items which are compatible with the user device; e.g., if the device used has not audio capabilities, the concrete document cannot contain audio files. In Figure 5 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 its format (F) is among the supported ones (SF).

2. Select from the list of the retrieved items those compatible with the user preferences, e.g., the language.

3. Consider 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 library or in a room which requires silence, audio files cannot be delivered unless the device is provided with headsets (function *adapt_env* of Figure 5).

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 at least one instance exists for the specified context; oth-

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)).
```

Figure 5. A sample of media features, device properties and adaptation rules.

erwise the instantiation fails, i.e., no concrete document can be generated for that context.

5. Process the optional virtual components only if at least one concrete instance exists for the specified context.
6. If context-selectable instances exist, make a further selection through some priority mechanism, intelligent assistant, heuristic evaluation, or by asking the user[3].

As an example of adaptation we consider the meteorological forecast document introduced in Figure 2 adapted to a PDA (situation c). We assume the user is in a library and does not wear headsets. Considering the structure depicted in Figure 3, the system first selects the media items compatible with the PDA device (step 1), i.e., all the text files, the satellite image (*sat-map*) and the small forecast map (*smallmap*); the animation and the big image are not selected because they do not fit the size of the PDA screen. Since the library requires silence and the user has no headsets, the system discards the audio comments (step 3).

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

The document obtained so far contains two instances of *context-selectable* items (see Section 5), i.e., the text files, which must be uniquely selected before playback. 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 details and can fit the PDA display.

Once the system has retrieved all the compatible media items, the instantiator builds the XML file of the concrete document by instantiating the dynamic structure of the presentation. It retains the synchronization relations between the virtual media and virtual composites which have produced concrete instances of their content; the relations are bound to the concrete media or composites instead of the virtual ones, generating the final document instance as in case (c) of Figure 2

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 [8] 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 allowed 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 4, 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 [2].

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.

Acknowledgment

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

References

[1] P. Brusilovsky. Methods and techniques of adaptive hypermedia. *User Modeling and User Adapted Interaction*, 6(2–3):87–129, 1996.

[2] A. Celentano, O. Gaggi, and M. Sapino. Retrieval in Multimedia Presentations. *Multimedia Systems*, 10(1):72–82, 2004.

[3] G. Chen and D. Kotz. A survey of context-aware mobile computing. Technical Report TR2000-381, Dartmouth College, Department of Computer Science, 2000.

[4] A. K. Dey. Understanding and Using Context. *Personal Ubiquitous Computing*, 5(1):4–7, 2001.

[5] J. Fink, A. Kobsa, and A. Nill. Adaptable and adaptive information provision for all users, including disabled and elderly people. *The New Review of Hypermedia e Multimedia*, 4:163–188, 1998.

[6] O. Gaggi and A. Celentano. Modelling Synchronized Hypermedia Presentations. *Multimedia Tools and Applications*, in press, Preliminary version: Techn. Rep. CS-2002-11, Dipartimento di Informatica, Università Ca' Foscari di Venezia, 2002, http:/www.dsi.unive.it/˜auce/docs/cs0211.pdf.

[7] A. Harter, A. Hopper, P. Steggles, A. Ward, and P. Webster. The anatomy of a context-aware application. In *Proc. 5th Annual ACM/IEEE Int'l Conf. on Mobile Computing and Networking*, pages 59–68, Seattle, Washington, August 1999.

[8] V. D. J. Euzenat, N. Layaïda. A Semantic Framework for Multimedia Document Adaptation. In *Proc. 18th Int. Joint Conf. on Artificial Intelligence*, pages 31–36, 2003.

[9] G. Klyne, F. Reynolds, C. Woodrow, H. Ohto, J. Hjelm, M. H. Butler, and L. Tran. Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies 1.0. W3C Recommendation, 2004.

[10] M. Kraus, F. Bry, and K. Kitagawa. 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)*, December 2003.

[11] T. Lemlouma and N. Layaïda. Adapted content delivery for different contexts. In *IEEE Symposium on Applications and the Internet (SAINT 2003)*, pages 190–199, 2003.

[12] T. Lemlouma and N. Layaïda. Context-aware adaptation for mobile devices. In *Proc. IEEE Int. Conf. on Mobile Data Management*, pages 106–111, 2004.

[13] A. Ranganathan and R. H. Campbell. An infrastructure for context-awareness based on first order logic. *Personal Ubiquitous Computing*, 7(6):353–364, 2003.

[14] B. N. Schilit, N. Adams, and R. Want. Context-aware computing applications. In *Proc. Workshop on Mobile Computing Systems and Applications*. IEEE, December 1994.

[15] A. Schmidt, M. Beigl, and H.-W. Gellersen. There is more to context than location. *Computers and Graphics*, 23, 1999.

[16] M. Schneider-Hufschmidt, T. Kühme, and U. M. U. (eds.). *Adaptive user interfaces: Principles and practice*. Elsevier Science, Amsterdam, 1993.

[17] L. Villard. Authoring transformations by direct manipulation for adaptable multimedia presentations. In *Proc. 2001 ACM Symposium on Document engineering*, pages 125–134. ACM Press, 2001.

[18] L. Villard, C. Roisin, and N. Layaïda. A XML-based multimedia document processing model for content adaptation. In *Proc. of Digital Documents and Electronic Publishing (DDEP00)*, volume 2023 of *LNCS*, Munich, Germany, September 2000. Springer.

[19] M. Weiser and J. S. Brown. The coming age of calm technology. In *Beyond Calculation: The Next Fifty Years of Computing*. Springer-Verlag, 1997.

[20] S. Wirag. Modeling of adaptable multimedia documents. In *Proc. Int. Workshop on Interactive Distributed Multimedia Systems and Telecommunication Services (IDMS'97)*, pages 220–230, 1997.