A CONTEXT-AWARE FRAMEWORK FOR MULTIMODAL DOCUMENT DATABASES

AUGUSTO CELENTANO and OMBRETTA GAGGI

Dipartimento di Informatica, Università Ca' Foscari di Venezia
via Torino 155, 30172 Mestre (VE), Italy
email: {auce,gaggi}@dsi.unive.it

We present a framework for the design and management of multimodal document databases: virtual documents describe the different facets under which information can be delivered and presented, in terms of media, content, and presentation format. Concrete documents are composed and instantiated from virtual ones by aggregating and synchronizing media instances based on the user context, i.e., the situation, the environment, the devices and communication resources available.

1. Introduction

In the framework of current technology for information processing and delivery, multimedia database management systems cannot limit their scope to the management and integration of different media according to predefined static schema or recurring templates. Rather, the growth of information service providers, and a wide spread of different features in delivery networks performance (e.g., Ethernet vs. Wi-Fi vs. GPRS), in user devices (desktop and notebook vs. PDA vs. smart phones), in user context (e.g., location, profile and environment variants), and in user interfaces (e.g., auditory vs. visual vs. haptic interfaces) make compelling demand of richer models and management systems for documents characterized by three key terms: multimedia, multichannel and multimodal[3,7]. In the scope of this paper we'll assume the following meaning for such terms:

- *multimedia*: a continuous document (also called a *multimedia presentation*) made of several synchronized media items (component documents), delivered to the user in a timely scheduled sequence, with several degrees of user control ranging from simple VCR style commands up to free hypermedia navigation;
- *multichannel*: a document which can be delivered, in different

formats and with different content versions and levels of details, through different communication and delivery channels, such as wide band networks for desktop screens or GPRS networks for portable phone displays;

- *multimodal*: a document whose content belongs to several communication codes and can be 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. For example, a multimodal document has a multimedia content, and is also multichannel since the different modalities can be delivered by different channels. For simplicity, in the paper the term *multimodal document* will be used to subsume also the multimedia and multichannel properties.

Multimodal documents can be effectively employed in many domains, and in many cases they are already successfully employed; among them, cultural heritage, virtual fairs, on-the-job training, entertainment and news are the most promising areas of exploitation[10,12,13].

In this paper we introduce a context-aware framework for multimodal document databases. Context-awareness means that the system is able to capture the *context* in which the document has to be delivered and to properly select the component media, the presentation modalities and the delivery channels.

A lot of work has been done in the area of context-awareness in recent years. Chen and Kotz[6] describe a wide range of context-aware applications. A simple system is the Active Badge System[19], that allows a receptionist to forwards a telephone call to the user's nearest phone, taking care of the user's current activity (e.g., calls are not forwarded during a meeting). A more complex application is the Office Assistant[21], an agent which interacts with visitors at office door and manages the office owner's schedule.

Among the many definitions of context applied to human-computer interaction we appreciate the one of Dey[8] for its wide scope: *"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"*. According to other authors[11,15,16], this information includes several facets among which we can consider of primary importance:

- the user context: typically, location, time and mobility, but also profile, situation and history;

- the environment context: e.g., indoor/outdoor, light/darkness, sound/silence;
- the communication context: e.g., wide/narrow band, plugged/ wireless, continuous/discontinuous;
- the device context: screen size and resolution, audio fidelity, input devices, user attention, and so on.

Other features could be of interest for specific application fields; e.g., heavy traffic prevents a driving user from receiving unanticipated visual messages on the navigator screen, while audio messages can be received because they are less distracting.

Mobile applications are deeply investigated since the situation of the user can change continuously, therefore automatic acquisition of context information is a key issue. Schmidt et al.[16] focus on the use of sensors to get context information, in a prototype which improves the user interface of mobile devices up to details like landscape or portrait device orientation.

A model of context must be capable of inferring complex information about the user situation from simple data. Ranganathan et al.[14] propose a model based on first order logic predicate calculus. Contextual information is expressed in term of predicates and complex rules on which the model allows to do automated inductive and deductive reasoning. Villard et al.[18] describe a general framework for context-adaptable hypermedia document production. A multimedia document is encoded independently from its presentation using an XML DTD. With the use of a Java implementation and of a XSLT processor the document is adapted to presentation device and to user preferences.

In the document management domain, 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[20].

2. Multimodal documents modelling

The design of a multimodal document database faces three issues related to document modelling: the static and the dynamic architecture of documents, and the context dependencies. The static architecture defines the structure of the document's contents, usually hierarchical. The dynamic architecture describes the media behaviour in time and in response to user interaction. The context dependencies link the document components to the context dimensions allowing the system to deliver the proper combination of media, content and presentation through the proper channels.

Several document models provide the foundations for the above issues. We ground our presentation on a multimedia document model we have defined in previous works[4,9] and used in several application contexts, which is here applied to multimodality and context-awareness.

Static architecture

Basically, a multimodal document is a collection of *atomic* and *composite* media items. Even if the elements of the collection can be documents at all effects, we shall use the term *item* for them, using the word *document* for denoting the set of items delivered to the user as a whole.

An *atomic* item is a self contained chunk of information, and is materialized in a database record or in a media file external to the database. Continuous media, like audio and video, have a temporal behaviour, therefore they can act as triggers in the dynamic architecture. Static media, like text and images, can be constrained to follow the dynamic behaviour 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.

Each item requires a set of resources for its delivery, i.e., for transmitting and playing it back on the user device. We do not distinguish here between communication resources and playback resources, because the difference is not relevant in the context of this paper. We generally call *channel* a resource or a set of related resources needed by a multimedia information item to be delivered in a specified context.

Since a multimodal 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 and channels than effectively used in each context. The component items could be delivered in the same way to different channels (e.g., a brief text), or can be specific to a channel (e.g., a video at some resolution).

Dynamic architecture

Temporal constraints define the evolution of the multimodal document in time, i.e., which media items are activated in sequence or in parallel, depending on mutual synchronization relationships. Several multimedia models discussed in literature could be used at this aim (see for example Bertino et al.[2]), since at the database management level the mapping between context and document dynamics is relevant, but not the actual multimedia model used, to which the DBMS would eventually conform.

In order to consider the interaction context of the user, the model must be able to handle not only the "natural" behaviour of a multimodal composed document, but also the variants in its dynamics introduced by the user. The model we have defined is well suited for this purpose, since it describes the synchronization relationships among the components of a multimedia document in an event-based fashion.

Synchronization between different components of a multimedia document is described by a small number of synchronization relations, which define media object reactions to some events like starting and ending of continuous media items which rule the presentation timing, and user actions on media. In order to settle a simple background for the discussion of context-awareness, we limit our discussion to the two relations describing basic parallel and sequential execution of media items. The forthcoming discussion therefore can as well be applied to documents described by other models, like the Allen temporal relations[1] and SMIL[17].

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. If item a is stopped before its natural end, the relation has no effect.

The timing relationships induced by $\Leftrightarrow$ and $\Rightarrow$ are not equivalent to parallel and sequential execution of media items as described, for example, by SMIL. Subtle differences exist[9] which are not relevant in the context of this paper.

Since static media have no timing properties, in our model their time length is in principle infinite, and their actual duration is ruled by synchronizing them with continuous media or with user actions. In order to allow a user to control media activation at his/her pace a special item is introduced, acting as a button: it is a continuous media item with a variable time length, which ends when the user activates it, e.g., with a mouse click.

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

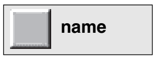
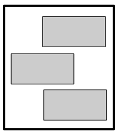
	An atomic item; the symbol describes the media type: video, audio, text, and image.
	A button for user control of media timing.
	Composite media items are represented by thick rectangles enclosing the component items.
⇔ ⇒	Synchronization relationships label the arrows connecting the involved media items.

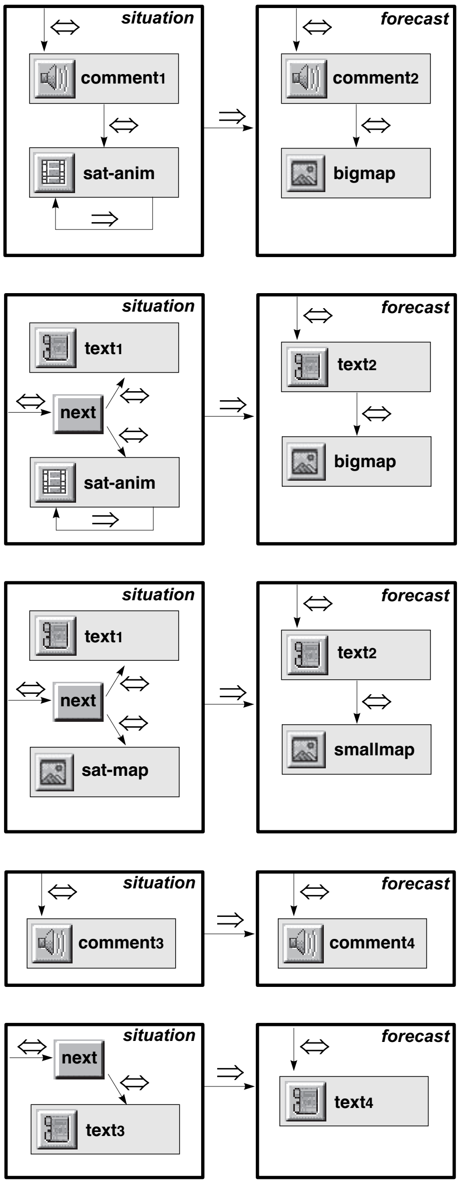
Figure 1. The graphical symbols used to represent synchronized multimedia documents.

Context constraints

Context constraints describe how media items are selected for building an instance of a multimodal document. 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 3.

A multimodal database is a collection of *virtual documents* which are made of *virtual components*, i.e., composite and atomic items, according to a structure independent from the context. Each virtual component is a collection of *instances*, which are elements on which context constraints are associated. The instantiation of a virtual document into a concrete document consists in the identification of the document components and, for each of them, in the selection of the proper instances compatible with the given context.

Figure 2 illustrates five versions of a document presenting a meteo forecast. Each version is contextualized 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). A virtual multimodal document collects into a unique structure all the variants with the associated context information (Figure 3).



(1) A multimodal document with full audio and animated video, suited for a desktop computer; the satellite image animation loops as long as the audio describes it; when audio ends, a forecast map is displayed and described by another audio comment.

(2) In this document the audio comment is substituted by a text, making the presentation suitable for silent environments; the duration of the text is set by the user who reads it and manually advances to the second part; the forecast is presented as in case 1.

(3) The PDA version of the document replaces the satellite animation with a small image; the forecast map is also a small image. As in case 2, user controls timing by advancing manually from the first part to the second.

(4) An audio only document to be delivered to a cellular phone.

(5) A text only document for delivery on a cellular phone as a sequence of short messages (SMS). User advanced from one message to the next.

Figure 2. A meteo forecast document for different contexts: a description of the meteorological situation is followed by the forecast. The presentation structure is constant, but different media and different synchronization schemas are used in different contexts.

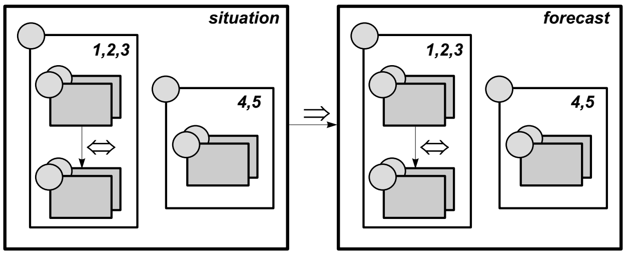


Figure 3. A virtual document containing cases 1–5 of Figure 2. Composites collect multiple elements for different contexts; the numbers in the upper right corner of the composites refer to the cases of Figure 2. The instances are selected by interpreting the context information associated to them, represented by circles in the figure.

A multimodal database is also accompanied by the definition of the *context* in which it operates. A database context is a set of features $\{f_1, f_2, \ldots, f_n\}$, each feature describing a property of a device, of the user, of the environment, etc., that can affect the document instantiation. A document context is a collection of pairs $\langle f, v \rangle$ where f is a context feature and v is an instance of that feature (a value), e.g., $\langle \mathit{displaysize}, \mathit{large} \rangle$ or $\langle \mathit{video}, \mathit{no} \rangle$.

Contexts are stored in the database as usual relational tables. We do not elaborate further here, noting however that contexts are structured along classes of features, possibly hierarchically defined, which belong to different domains like the user profile, the user device, and so on.

3. Context-awareness in document modelling

A component of a virtual document can be mandatory or optional according to its role in the document semantics. A mandatory component must always be present in the document for any context; if it is missing, the document is *incomplete* and cannot be delivered. An optional component can be present or not, i.e., it can be delivered or not, according to its compatibility with the specific context attributes during the concrete document instantiation.

The properties of a mandatory component are the following ones: for each value or set of values of a context attribute an instance must be defined; conversely, an instance is associated to a list of context attributes, and for each attribute to a list of values which define the compatible contexts. At instantiation time if an appropriate context is not available, instantiation is not possible and the virtual document to which it belongs cannot be delivered.

An optional component can be instantiated (hence delivered) or not without preventing the document from being understandable and useful; the semantics and correctness of the document are not affected by the component presence. The delivery can be a system choice, according to context compatibility, but also a user choice according to a selection made at document request time; delivery is in any case bound to the availability of an instance compatible with the current context.

A *context-independent* component exists in only one instance, which is compatible in principles with all contexts, even if it cannot be delivered on some channels (e.g., an audio device cannot play a written text, but a written text can be displayed in the same way independently from other types of context. At instantiation time the system does not perform any selection but simply picks up the item or not depending on the available channel.

A component is *context-dependent* when it exists in several instances, one for each context or list of contexts; in some contexts the content can be void[a]. For example, a video only component could be void in a mobile phone context, while in a desktop or PDA context could exist in several resolutions and frame speeds. The selection is performed by the system.

A *context-selectable* component exists in several instances, possibly more than one for some contexts. For example, a text at several degrees of detail, an image at several resolutions, alternative spoken/written versions of the same text. The selection is partly performed by the system (by picking only the instances compatible with the current context), and partly by the user, who selects among the alternatives the one best suited to his/her taste. An intelligent agent approach could give support to this issue, but we do not elaborate on this issue here.

A *concrete document* is an instance of a virtual document for a specific context or combination of contexts.

4. Multimodal documents instantiation

As we have seen in Section 2 the instantiation of a concrete document from a virtual one consists in the identification of the relevant virtual components and, for each of them, in the selection of the instances according to the context. The identification of the components is trivial if the whole document

[a] A void instance is different from an optional component, since optionality is referred to a semantic role in the document, while the instance is related to the possibility of delivering an item to the user within the current contexts.

has to be delivered, since they are listed in the document static structure. However, the delivery of a whole document is not the most frequent case, due to the limitations of some devices or communication channels, and to the context variations in some environments like the mobile ones, 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 a long download. Therefore, delivering a document generally requires 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 document playback.

The identification of the relevant items of each part is in fact a retrieval task: the database is queried for resolving the context-dependencies, based on information extracted by the analysis of the static and the dynamic relationships among the virtual document components. For space reasons we do not discuss here such issues, referring the reader to a previous work by the same authors[5].

In database terms the virtual document instantiation is a view building operation that requires several steps:

- identify the needed set of components for delivering a document section; i.e., given the main component of the section, identify the media items bound to it by a synchronization relationship, in order to build up a coherent and complete segment for the user;
- for each mandatory virtual components, check that one or more instances exist for the specified context; if some of them do not exist the instantiation fails;
- process the optional virtual components only if at least one concrete instance exists for the specified context;
- select the set of concrete documents compatible with the specified context;
- if context-selectable instances exist, build appropriate combinations by asking the user or through some intelligent assistant, heuristic, or other[b];

[b] The details are not relevant since the instances are assumed functionally equivalent for

- deliver the concrete document to the user according to the defined dynamics and synchronization constraints.

5. 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 to components context information.

Among the issues that deserve further investigation, the consistency of a complex document under different context conditions is of paramount 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, who should not be involved in explicit selection operations.

Consistency can be approached by attaching rules to the components (both virtual and concrete) that describe mutual consistency relationships, as much as in a traditional database referential integrity defines mutual constraints between database records.

References

1. J. F. Allen. Maintaining knowledge about temporal intervals. *Comm. ACM*, 26(11):832–843, November 1983.
2. E. Bertino and E. Ferrari. Temporal Synchronization Models for Multimedia Data. *IEEE Transactions on Knowledge and Data Engineering*, 10(4):612–631, July/August 1998.
3. M. M. Blattner and E. P. Glinert. Multimodal integration. *IEEE Multimedia*, 3(4):14–24, 1996.
4. A. Celentano and O. Gaggi. Template-based generation of multimedia presentations. *International Journal of Software Engineering and Knowledge Engineering*, 13(4):419–445, 2003.
5. A. Celentano, O. Gaggi, and M.L. Sapino. Retrieving Consistent Multimedia Presentation Fragments. In *Workshop on Multimedia Information Systems (MIS 2002)*, pages 146–154, Tempe, Arizona, USA, November 2002.
6. G. Chen and D. Kotz. A survey of context-aware mobile computing. Technical Report TR2000-381, Dartmouth College, Department of Computer Science, 2000.
7. M. Delato, A. Martelli, M. Martelli, V. Mascardi, and A. Verri. A multimedia, multichannel and personalized news provider. In G. Ventre and R. Canonico,

that context.

editors, *Proc. of the First Int. Workshop on Multimedia Interactive Protocols and Systems, MIPS 2003*, pages 388–399. Springer-Verlag, 2003. LNCS 2899.

8. Anind K. Dey. Understanding and Using Context. *Personal Ubiquitous Computing*, 5(1):4–7, 2001.
9. O. Gaggi and A. Celentano. Modelling Synchronized Hypermedia Presentations. *Multimedia Tools and Applications, Kluwer Publ. Co.*, in press, 2004. Preliminary version: Technical Report CS-2001-11, Dipartimento di Informatica, Università Ca' Foscari di Venezia, 2002, http:/www.dsi.unive.it/˜auce/docs/cs0211.pdf.
10. S. Izadi, M. Fraser, M. Flintham S. Benford, and C. Greenhalgh. Citywide: Supporting interactive digital experiences. In Dunlop and Brewster, editors, *Mobile HCI 01 - Proceedings of the Third International Workshop on Human-Computer Interaction with Mobile Devices*, pages 41–46, November 2001.
11. K. Mitchell N. Davies, K. Cheverst and Alon Efrat. Using and determining location in a context-sensitive tour guide. *IEEE Computer*, 34(8):35–41, 2001.
12. F. Pittarello. Multi sensory 3d tours for cultural heritage: the palazzo grassi experience. In *Proc. of ICHIM2001 - Cultural Heritage and Technologies in the 3rd Millennium*, 2001.
13. I. Rakkolainen and T. Vainio. A 3d city info for mobile users. *Computers & Graphics, Special Issue on Multimedia Appliances*, 25(4):619–625, 2001.
14. Anand Ranganathan and Roy H. Campbell. An infrastructure for context-awareness based on first order logic. *Personal Ubiquitous Computing*, 7(6):353–364, 2003.
15. B. N. Schilit, N. Adams, and R. Want. Context-aware computing applications. In *Proc. Workshop on Mobile Computing Systems and Applications.* IEEE, December 1994.
16. A. Schmidt, M. Beigl, and Hans-W. Gellersen. There is more to context than location. *Computers and Graphics*, 23, 1999.
17. Synchronized Multimedia Working Group of W3C. Synchronized Multimedia Integration Language (SMIL) 2.0 Specification, August 2001.
18. L. Villard, C. Roisin, and N. Layada. A XML-based multimedia document processing model for content adaptation. In *Proceedings of Digital Documents and Electronic Publishing (DDEP00)*, volume 2023 of *Lecture Notes in Computer Science*, Munich, Germany, September 2000. Springer.
19. R. Want, A. Hopper, V. Falcão, and J. Gibbons. The active badge location system. *ACM Trans. Inf. Syst.*, 10(1):91–102, 1992.
20. M. Weiser and J. Seely Brown. The coming age of calm technology. In *Beyond Calculation: The Next Fifty Years of Computing*. Springer-Verlag, 1997.
21. H. Yan and T. Selker. Context-aware office assistant. In *Proceedings of the 5th international conference on Intelligent user interfaces*, pages 276–279. ACM Press, 2000.