Presentation Scheduling of Multimedia Objects and Its Impact on Network and Operating System Support

PETRA HOEPNER

GESELLSCHAFT FÜR MATHEMATIK UND DATENVERARBEITUNG (GMD)
FORSCHUNGSZENTRUM FÜR OFFENE KOMMUNIKATIONSSYSTEME (FOKUS)
HARDENBERGPLATZ 2
D - 1000 BERLIN 12

The presentation of a multimedia object is concerned with the conveyance of information for a human user. The information conveyed depends not only on the contents presented, but also on their layout in time and space, i.e. on a specific presentation behaviour. The specification of the presentation behaviour in a system- and application-independent way provides for a compatible presentation of the multimedia object at different locations of a distributed, heterogeneous system. The paper concentrates on the temporal aspects of an abstract presentation behaviour description, i.e. the synchronization and control of temporally related presentation actions. The operating system and network support required for the execution of a presentation schedule is outlined.

1 Introduction

The *presentation of multimedia objects* is concerned with the conveyance of information for a human user. The information conveyed is not based solely on common representation types for the presentation [1] (e.g. character text, geometric graphics, moving pictures/images, audio/speech, raster graphics/images), but also on the resources/devices involved, on the transformation of the application-specific representation types (e.g. product definition data, trade data, animation) into a human perceptible form and on the control of spatial, temporal and hypermedia (hyperlinks) relations. In addition, an active participation of the user in the presentation process (interaction) must be possible.

Regarding different multimedia information/data models (e.g. [2], [3], [4], [5], [6], [7], [8], [9]) the gradation of presentation behaviour specification varies from the prescription of only the basic information/data objects associated with some content to composite information/data objects imposing also spatial, temporal and/or hypermedia constraints on the presentation of the information/data. The description of the intended presentation behaviour is required by diverse multimedia applications such as multimedia mail systems [10], educational software [11], etc.

If a specific presentation layout in a distributed heterogeneous environment is intended, an exchange format for multimedia information/data has to provide for content exchange as well as for conveying structuring information. Spatial, temporal and/or hypermedia structuring capabilities can thereby be integrated in the information/data model used, such as proposed for the integration of temporal relations into the 'Open Document Architecture (ODA)' [12], [13], can be defined separately in, for example, a script or can be internal to the data as, for example, in the data format of DVI.

This paper is mainly concerned with the specification of the temporal aspects of the presentation behaviour and the requirements imposed on system behaviour.

Temporal relations can be generated at the creation time of data (termed 'live synchronization' [12]) or set up artificially for stored data (termed 'synthetic synchronization' [14]). The temporal relations for both, live and synthetic synchronization, can be specified by a *synchronization rule.* The synchronization rule determines the goal of synchronization [15]; in the following context this goal is the temporal ordering and control of presentation actions and interactions termed *presentation scheduling.* Detaching presentation scheduling and its specification from any specific application, information/data format or concrete synchronization or scheduling mechanism, advocates the so-called 'separation of concerns', [16] proposed for the specification of hard real-time systems to achieve advantages such as the flexibility to change and the detection and correction of errors.

Presentation schedules of multimedia objects to be performed in heterogenous, distributed environments do strongly rely upon network and operating system support. The general requirements for such a support will be outlined.

2 Presentation Schedule

The presentation of a multimedia object comprises a set of temporally related actions and interactions, as defined by the originator, user or some other instance. Different approaches for the modelling and the specification of temporal relations and their synchronization exist, such as [3], [7], [12], [13], [17], [18].

An extraction of the basic concepts provided within these approaches leads to the following modelling requirements regarding the synchronization and control of the presentation:

- *Actions*. The presentation of a multimedia objects can be decomposed into a set of actions, such as present picture X, play audio comment Y, show menu Z, get menu selection.
- *Temporal Relations*. Actions are temporally related to each other thereby defining a specific presentation order, such as sequential, parallel.
- *Reconciliation*. The performance of an action may be modified or adapted to accomplish the presentation order specified, such as clip, fill, repeat.
- *Control*. The performance of one or more actions may be influenced by external control functions, such as stop, pause, resume.

The specification of a *presentation schedule* takes into consideration the requirements listed above. Interpreting or compiling a presentation schedule leads to a specific arrangement of processes in a distributed system which behave according to the specification.

In [13] a synchronization model for the presentation was described, based on actions and their temporal relations. An *action* is generally the "representation of something which happens" as defined in the 'Reference Model of Open Distributed Processing (ODP-RM)' [19]. A presentation action consists of events which construct a time interval. Some events are significant for the synchronization of actions, the *synchronization events*. Start- and endpoints of actions are synchronization events. Depending on the (non)existence of synchronization events between the start- and endpoint of an action, 'atomic actions' and 'composed actions' can be differentiated. Composed actions are actions that are composed from atomic actions or composed actions. The participating actions of a composed action have to be synchronized to fulfill a defined order of execution. An extension of the model with reconciliation and control concepts for the specification of presentation schedules is introduced in the following.

The temporal extent of a presentation action can be defined by its inherent duration (e.g. 'play video X') or can be delimited externally (e.g. 'show picture P one minute', 'show picture P until the end of the music M'). This temporal extent is termed *presentation frame* in the sequel. The extent of a presentation frame can be defined internally or externally, for example, by end of the contained actions, by a specified duration with respect to the start of the presentation frame (this can be also an absolute duration) or by external events such as the start of another presentation frame or a termination event issued by the user at the control interface.

An action can be reconciled to the extent of the presentation frame by so-called extent reconciliation strategies. Strategies to accomplish the extent reconciliation are (based on HyTime [4]):

Alignment A specific alignment strategy for the action can be specified with respect to the presentation frame extent (e.g. fading, stretching, shrinking, repeat number etc.) (prior to vamping, filling, clipping or center).

Vamping The action is repeated until the presentation frame extent is reached.

Filling The rest of the extent is filled with a 'malleable action', i.e. one that can be scaled arbitrarily without unacceptable distortion, such as pause, or with an alternative action (concept of 'restricted blocking' introduced in [17]).

Replacement Potential replacement actions are defined for example for the best fit.

Clipping If an action is larger than the presentation frame extent, it will be clipped.

Center If the presentation frame extent is known, the action will be centered within the extent (e.g. accompanied by a 'filling' strategy).

Offset The action is shifted 'offset' time units from the startpoint of the presentation frame (e.g. accompanied by a 'filling' strategy).

Duration The action is performed for a specified 'duration'. This is especially useful for actions with no inherent timing.

A presentation frame may not only delimit the extent of one action, but also the extent of two or more actions. Therefore a temporal relation must be associated with the presentation frame, which defines the synchronization strategy for the contained actions. In Section 3 different types of presentation frames are introduced, in each case prescribing a temporal relation (based on the concepts described in [19] and [13]).

The termination of a presentation frame ends all contained actions, i.e. the actions are no longer performed. Note, that 'not performing' an action logically can imply physically a negating action, such as presenting a picture and removing the picture.

With respect to a specific presentation schedule a presentation frame is identified uniquely by a 'presentation-frame-id'. Actions contained in a presentation frame can be referenced by the following notation 'presentation-frame-id.action-id'.

3 Presentation Frame Types

3.1 Sequentializer

A *Sequentializer* contains one or more actions which are performed in sequence (ODP-term: chain of actions). The start of the sequentializer starts the first action in sequence. The end of the sequentializer may be defined by the end of the last action in the sequence or by a specific presentation frame extent. Additional constraints for the sequential performance can be specified, that is continuation or non-continuation. Continuation requires an exact sequential performance, i.e. the endpoint of one action corresponds to startpoint of the next action in sequence. Non-continuation does not impose such a rigorous constraint, i.e. sequential actions start as soon as possible one after each other, for example, between pictures there may be a delay for the time the picture is prepared for presentation.

EXAMPLE:

```
FRAME       presentation_frame_name
TYPE        sequentializer
START_EVENT event_from_other_frame
END_EVENT   end_presentation_frame_name = end_action_C
CONSTRAINT  continuation
ACTION      A  with  RECONCILIATION duration(5)
ACTION      B
ACTION      C
```

The presentation frame 'presentation frame name' defines the sequential performance of the actions 'A', then 'B', then 'C'. The start of the presentation frame is signalled by the event 'event from other frame'. The end of the presentation frame is signalled to the environment by event 'end presentation frame name', which is internally determined by the end of action 'C'. The endpoint of one action is exactly the startpoint of the next action. Action 'A' lasts 5 time units (e.g. a picture shown for 5 minutes). For actions 'B' and 'C' no duration is specified, which means either that the actions perform for a certain time, such as audio and video actions, or that they are terminated externally (see Section 4).

3.2 Parallelizer

A *Parallelizer* contains two or more actions. All actions start relative to the beginning of the parallelizer (similar to the ODP-term: forking action). The end of the presentation frame can be defined by (1) a specific action, i.e. a master action, by (2) the end of a non-deterministic action, i.e. the first, second, ..., last action in time, or by (3) a specific presentation frame extent. Extent reconciliation strategies are possible, for example, all actions are clipped if the master action terminates.

Tight synchronization requirements, such as lip synchronization between an audio and a video action, may be modelled by a fine granularity of parallelizer presentation frames containing atomic actions on audio/video frame basis (as modelled in [7] with Petri-Nets).

EXAMPLE:

```
FRAME       presentation_frame_name
TYPE        parallelizer
START_EVENT event_from_other_frame
END_EVENT   end_presentation_frame_name = end_master
CONSTRAINT  master is A
ACTION      A
ACTION      B  with  RECONCILIATION alignment(shrink OR stretch)
ACTION      C  with  RECONCILIATION offset(2) AND clip
```

The presentation frame 'presentation frame name' defines the parallel performance of the actions 'A', 'B' and 'C'. The start of the presentation frame as signalled by the event 'event from other frame' starts the actions 'A' and 'B'; the action 'C' is shifted two time units from this startpoint. The end of the presentation frame is signalled to the environment by the event 'end presentation frame name', which is internally determined by the end of the master action 'A'. Action 'B' is exactly aligned to the presentation frame extent by shrinking or stretching its extent, the action 'C' is clipped, when the presentation frame ends.

3.3 Splitter

A *Splitter* contains two or more actions. All actions start relative to the beginning of the splitter. The difference from the parallelizer is, that more than one synchronization event can be issued (ODP-term: dividing action). A splitter therefore splits the presentation into independent branches, i.e. independent parts of the presentation, because different presentation frames may be scheduled to the various synchronization events. The presentation frame ends when all synchronization events have occurred. For example, two actions A and B are started at the same time. However, the end of the first action in time starts presentation frame X and the end of the last action in time starts presentation frame Y. Extent reconciliation strategies have to be chosen carefully because, for example, an aligning of endpoints may be contradictory to the semantics of this presentation frame type.

EXAMPLE:

```
FRAME       presentation_frame_name
TYPE        splitter
START_EVENT event_from_other_frame
SYNC_EVENT  sync_event_1 = end_first
            sync_event_2 = end_action_A
END_EVENT   end_presentation_frame = sync_event_1 AND sync_event_2
ACTION      A
ACTION      B  with  RECONCILIATION clip
```

The presentation frame 'presentation frame name' defines the parallel performance of the actions 'A' and 'B'. The start of the presentation frame as signalled by the event 'event from other frame' starts the actions 'A' and 'B'. The synchronization event 'sync event 1' is signalled to the environment as soon as the first action in time terminates (event 'end first'). Independently, the end of action 'A' is signalled to the environment by the event 'sync event 2'. Therefore two different cases of presentation ordering can be differentiated: (1) action 'B' is the shorter action in time, action 'A' is then continued until it terminates ('sync event 2' and 'end presentation frame' are issued simultaneously) and (2) action 'A' is the shorter action in time, all three synchronization events happen at the same time and action 'B' is clipped.

3.4 Combiner

A *Combiner* joins two or more independent branches of the presentation schedule (ODP-term: joining action). The start of the presentation frame therefore depends on more than one synchronization event, i.e. the synchronization events of two or more preceding presentation frames. A combiner may contain no action or one action. Ending the presentation frame however emits only one synchronization event.

EXAMPLE:

```
FRAME       presentation_frame_name
TYPE        combiner
START_EVENT end_frame_X AND end_frame_Y
END_EVENT   end_presentation_frame
ACTION      none
```

The presentation frame 'presentation frame name' ends as soon as the the presentation frames 'X' and 'Y' have ended.

3.5 Brancher

A *Brancher* is a special presentation frame that represents the possibility of several alternative futures (e.g. a 'choice' construct in programming languages). A possible presentation schedule may be regarded as a tree; only one branch of the tree will be selected and performed. The brancher always contains an action which has the ability to determine a selection value, necessary for the decision of the branch. As soon as the value is determined the brancher ends, i.e. the contained action is finished with the end of the presentation frame. Following presentation frames depend not only on the end of the brancher but also on the selection value this presentation frame determined. Start constraints for the following presentation frames have to be specified according to that rule. It is also possible to hide the value from the environment if the brancher is able to determine which presentation frame waits for which value and emits a value-specific synchronization event.

EXAMPLE:

```
FRAME       presentation_frame_name
TYPE        brancher
START_EVENT event_from_other_frame
END_EVENT   end_presentation_frame = end_action_A
VALUE       input_value    = choice of 1,2,3
```

```
            default_value = 1
ACTION      A  with  RECONCILIATION duration(20)
```

The start of the presentation frame as signalled by the event 'event from other frame' starts the action 'A'. The action 'A' determines a value (1, 2 or 3). The default value '1' is adopted if no input value was determined. Action 'A' either terminates as soon as the selection is done or after 20 time units.

4 Presentation Frame Interfaces

Presentation frames can be regarded as objects enforcing the above-specified behavioural constraints. The construction of a specific presentation schedule as well as the active participation of the user in the presentation requires the provision of interfaces by presentation frames allowing for different types of interactions, namely

- interactions between presentation frames
- interactions between a user and a presentation frame

The interactions between presentation frames are determined by the presentation schedule based on the specification of the synchronization events offered and accepted by the various presentation frames. These interactions have to be supported by the operating system.

The interactions a user can participate in depend on the interface that a presentation frame exports, i.e. the methods provided. These methods may either influence the presentation frame as a whole, i.e. the performance of all contained actions, or influence only the performance of a specific action. Nevertheless both types of interactions may influence the temporal behaviour of the presentation frame. For example, the operation 'stop' on a presentation frame will terminate the presentation frame, thereby terminating all contained actions. Performing the operation 'stop' on a specific action will terminate this action in the presentation frame; this, however, may influence other contained actions and also the presentation frame, if, for example, the action is a master action.

5 Impact on Network and Operating System Support

The above-described presentation schedule specification requires the distribution and instantaneous reaction (ideal case) to synchronization events. This concept is the basis of a "reactive system" (as introduced by Harel and Pnueli). A reactive system is based on a strong synchronism hypothesis: any reaction (processing) in response to an event occurrence is instantaneous, and any reaction is synchronous with the event that produced it. [20] considered distributed multimedia applications in general as a collection of interacting objects, organized around particular reactive objects, called "reactive kernels". Analogous to this, the real-time aspects (control and synchronization) of a multimedia presentation can be handled by *reactive presentation kernels*.

For the programming of reactive systems some formalisms have been proposed, such as ESTEREL [21] and Statecharts [22]. These formalisms are based on the concept of an instantaneous broadcast communication mechanism as, for example, modelled by finite-state machines. Synchronization events are therefore broadcast immediately, i.e. instantaneously. Based on the synchrony hypothesis, the distribution of synchronization events of presentation frames or actions takes "no time".

The concept of a reactive system leads to a problem regarding the distribution aspect of the presentation [20]. A distributed presentation system has to cope with asynchronous behaviour as introduced by communication objects. To enable synchronous reactions to synchronization events during the presentation, the timeliness of information/data availability has to be supported by real-time communication services. Performance requirements therefore have to be specified [23]. Based on these requirements, a metascheduler [24] may be used that reserves components guaranteeing a certain quality of service. As far as buffering capabilities are available at the destination (location of presentation) and the requested data is persistent, a 'work ahead' in transmitting presentation data is possible; if no buffering capabilities are available, startpoint-point synchronization and real-time transmission of presentation data is required.

The execution of a presentation schedule also has to take into account the allocation of resources and the temporal behaviour of presentation devices regarding their interface for the aquisition and restitution of continuous media data. Real-time scheduling strategies [25] have to be applied in dependance on the available resources and the processes to be scheduled.

As outlined, the synchronization and control of the presentation is not a local feature, but has to be supported at different architectural levels of an open, distributed system, examined for example in [14] and [26].

The enhancement of presentation schedules with features supporting the specification of performance requirements (for example Quality of Service (QOS) parameters) for the communication as well as with resource characteristics is a subject for further study.

6 Conclusion

The presentation of a multimedia object consists of temporally related presentation actions. The synchronization and control of these presentation actions, i.e. the temporal behaviour, is specified in a presentation schedule. Presentation actions are arranged in presentation frames; presentation frames are composed into a specific presentation schedule. Presentation frames are responsible for the reconciliation of actions to the extent defined and for the provision of interaction interfaces to other objects, for example, other presentation frames or the user.

The execution of a presentation schedule by a reactive presentation kernel supporting the instantaneous broadcasting of synchronization events was proposed. The adherence to the synchronization rules defined by the presentation schedule also imposes temporal requirements on the transmission of data streams in the communication system.

The presentation of multimedia objects in open, distributed systems requires the support of several services, such as communication services, retrieval services, transformation services. All these services have to offer certain qualities to handle real-time information/data transfer and processing. Further research has to be done in the evaluation and realization of presentation schedules regarding the quality characteristics of supporting services in a distributed application platform [27].

References

[1] "BERKOM Reference Model - Application-Oriented Layers", ed. GMD-FOKUS, DETECON Technisches Zentrum Berlin, Germany (May 1991), Version 3.2.

[2] ISO 8613, "Information Processing - Text and Office Systems - Office Document Architecture (ODA) and Interchange Format" (1989).

[3] ISO/IEC JTC1/SC2/WG12, "Coded Representation of Multimedia and Hypermedia Information, Working Document for the Future MHEG Standard", Version 3 (November 1990).

[4] ISO N2948, "Information Technology - Hypermedia/Time-based Structuring Language (HyTime) Committee Draft 10744" (April 1991).

[5] R. Steinmetz, R. Heite, J. Rückert, and B. Schöner, "Compound Multimedia Objects - Integration into Network and Operating System", in: *First International Workshop on Network and Operating System Support for Digital Audio and Video*, International Computer Science Institut (ICSI): TR-90-062 (November 1990).

[6] F. Oguet, C. Schwartz, F. Kretz, and M. Quere, "RAVI, A Proposed Standard for the Interchange of Audio/Visual Interactive Applications", *IEEE Journal on Selected Areas in Communications*, vol. 8, no. 3, pp. 428–436 (April 1990).

[7] T.D.C. Little and A. Ghafoor, "Synchronization and Storage Models for Multimedia Objects", *IEEE Journal on Selected Areas in Communications*, vol. 8, no. 3, pp. 413–427 (April 1990).

[8] G. Blakowski, "Konzeption für eine Sprache zur Beschreibung von Transport- und Darstellungseigenschaften multimedialer Objekte", in: *Telekommunikation und multimediale Anwendungen der Informatik*, ed. J.Encarnacao (Hrsg.), pp. 465–474, Proc. of the GI'91, October 14-18 1991, Darmstadt, Informatik-Fachberichte 293, Springer Verlag (1991).

[9] S. Lehert and E. Moeller (eds.), *Data and Information Modelling*, Proc. of the BERKOM Workshop, Annelsbach - Hoechst/Odenwald, July 9-13, 1990, GMD Report no. 196, R. Oldenbourg Verlag, München/Wien (1991).

[10] J.B. Postel, G.G. Finn, A.R. Katz, and J.K. Reynolds, "An Experimental Multimedia Mail System", *ACM Transactions on Office Information Systems*, vol. 6, no. 1, pp. 63–81 (January 1988).

[11] M.E. Hodges, R.M. Sasnett, and M.S. Ackerman, "A Construction Set for Multimedia Applications", *IEEE Software*, no. 1, pp. 37–43 (January 1989).

[12] R.G. Herrtwich and L. Delgrossi, "ODA-Based Data Modeling in Multimedia Systems", International Computer Science Institut (ICSI): TR-90-043 (1990).

[13] P. Hoepner, "Synchronizing the Presentation of Multimedia Objects - ODA Extensions -", *ACM SIGOIS Bulletin*, vol. 12, no. 1, pp. 19–32 (July 1991).

[14] T.D.C. Little and A. Ghafoor, "Network Considerations for Distributed Multimedia Object Composition and Communication", *IEEE Network Magazine*, vol. 4, no. 6, pp. 32–49 (November 1990).

[15] P.Hoepner, G.Schürmann, and K.-H.Weiss, "Synchronization, Isochronous and Anisochronous Communication", *BERKOM-Projekt: "Multi-Media-Dokumente im ISDN-B"*, DETECON, Technisches Zentrum Berlin, Voltastr. 5, 1000 Berlin 65, Germany (März 1991).

[16] S.R. Faulk and D.L. Parnas, "On Synchronization in Hard-Real-Time Systems", *Communications of the ACM*, vol. 31, no. 3, pp. 274–287 (March 1988).

[17] R. Steinmetz, "Synchronization Properties in Multimedia Systems", *IEEE Journal on Selected Areas in Communications*, vol. 8, no. 3, pp. 401–412 (April 1990).

[18] ISO/IEC JTC1/SC18 N2028, "Information on French Proposal on Audiovisual Interactive Applications" (September 1989).

[19] ISO/IEC JTC1/SC21 N6079 CD Text, "Basic Reference Model of Open Distributed Processing Part 2: Descriptive Model" (July 1991).

[20] L. Hazard, F. Horn, and J.-B. Stefani, "ISA Project - Notes on Architectural Support for Distributed Multimedia Applications", Centre National d'Etudes des Telecommunications (1991).

[21] G. Berry, P. Couronne, and G. Gonthier, "Synchronous Programming of Reactive Systems: An Introduction to Esterel", INRIA, Institut National de Recherche en Informatique et en Automatique, Technical Report No. 647 (1987).

[22] D. Harel, "Statecharts: a Visual Formalism for Complex Systems", *Science of Computer Programming*, vol. 8, no. 3 (June 1987).

[23] D. Ferrari, "Client Requirements for Real-Time Communication Services", *IEEE Communications Magazine*, vol. 28, no. 11, pp. 65–72 (November 1990).

[24] D.P. Anderson, "Meta-Scheduling for Distributed Continuous Media", University of California, Computer Science Division, Technical Report No. UCB/CSD 90/599 (October 1990).

[25] R.G. Herrtwich, "An Introduction to Real-Time Scheduling", International Computer Science Institut (ICSI): TR-90-035 (July 1990).

[26] C. Nicolaou, "An Architecture for Real-Time Multimedia Communication Systems", *IEEE Journal on Selected Areas in Communications*, vol. 8, no. 3, pp. 391–400 (April 1990).

[27] R. Popescu-Zeletin, V. Tschammer, and M. Tschichholz, "'Y' distributed application platform", *Computer Communications*, vol. 14, no. 6, pp. 366–374 (July/August 1991).