Position Statement on Multimedia Synchronization

Michael Wynblatt
State University of New York at Stony Brook
wynblatt@cs.sunysb.edu
May 19, 1995

In the area of multimedia synchronization, I am primarily interested in issues regarding the specification of synchronization constraints and properties as they relate to multimedia authoring. In this position statement, two of these issues which have not received much attention are discussed, followed by a summary of how the Layered Multimedia Data Model, developed at Stony Brook, addresses them. Note that the term *composition* is used to mean a complete, ready to play, set of multimedia data and specifications.

1 Alignment vs. Synchronization

There is some discrepancy in the literature as to what exactly is meant by synchronization, in the context of multimedia. Some models view synchronization as constraints which describe at what times streams begin, and end, relative to each other (e.g. [6], [3]). Others view synchronization as constraints which cause concurrent streams to advance in a closely coupled manner (e.g. [4]). These two kinds of constraints are sometimes distinguished as *coarse* vs. *fine grain* synchronization [5]. Some approaches have a unified model, treating the former as a special case of the latter (e.g. [2], [7]). But they are not the same, and should be modeled differently.

The two types of constraints have different roles in the multimedia authoring process. Specification of relative ordering constraints, which we might refer to as *temporal alignment* rather than synchronization, is primarily a conceptual task, squarely in the domain of the multimedia author. For example, the determination of whether a set of video clips will only be shown concurrently or sequentially, or if the latter, in what sequence they should be shown, is mainly driven by high level design goals. These decisions can be made by the multimedia author, who understands the subject matter of the composition, but who may or may not be familiar with the technical issues of multimedia.

Conversely, specification of the second type of constraints, which we could refer to as *stream synchronization*, is more an issue of quality of service than conceptual design. Since the ideal synchronization between two streams cannot always be guaranteed, it necessary for a developer to specify what constraints are desired, and what compromises are acceptable (tolerance and resynchronization methods). Determination of these specifications is more likely to be decided based on quality and performance concerns than on conceptual design. For many ¡compositions¿, specification of stream synchronization constraints might even be automated, with no direct input from the developer at all.

Further, constraint specifications must be interpreted by the system in a way which generates correct playback. Although this is not usually difficult for temporal alignment, enforcing stream synchronization constraints can be quite demanding on the system. Stream synchronization constraints should therefore be specified in a way which maps to an effective model of playback.

While it is clear that both kinds of specification are necessary in a complete multimedia temporal model, we have proposed that it is useful to distinguish between them, and specify them in different ways [8]. Both are important, but have different requirements, and by modeling them separately, a more useful model of each is possible. For example, temporal alignment should be modeled in a user-oriented way, so that authors can easily relate streams in the order they choose. Stream synchronization should be modeled in a system-friendly way, so that efficiency concerns can be addressed at playback time, and automated specification may be possible.

It is worth noting that at least one group views synchronization as part of the more general problem of inter-object communication [1]. Although this is a valid viewpoint, it might be argued that the issues of temporal alignment and stream synchronization are significant enough in multimedia applications to command special consideration and modeling techniques.

2 Granularity vs. Strictness

Two important parts of the specification of stream synchronization are the *granularity* of synchronization and the *strictness* of synchronization [8]. A complete multimedia model must describe them both. Granularity represents the degree to which to streams must be coupled during playback. For example, motion video with speaking characters must be synchronized to the audio track many times per second, while simpler animations with audio may only require synchronization every few seconds. An automated slide-show need only synchronize with audio at the beginning of each slide.

The strictness of synchronization describes how synchronization constraints are interpreted. In any of the examples above, the desired degree of synchronization may not be available on a given system. The developer should be able to specify the degree of *tolerance* [8, 4] which the playback system must allow, and how to recover in case the constraints are violated.

These two specifications represent different parts of the authoring process. The granularity of synchronization is determined mainly by the content and types of the component data objects. Such constraints are part of the temporal structure of a composition, and are independent of any particular way of presenting the structure. Strictness specifications, on the other hand, may vary from platform to platform, as some platforms will have limited capability. Different presentations of a structure may require different strictness. Strictness specifications are not part of the temporal structure itself, but are an interpretation of that structure for a given presentation.

A model of synchronization should describe both granularity and strictness, and it should reflect the different roles they play in the authoring process. Where granularity of synchronization should be associated with the temporal structure of a composition, strictness should be associated with particular presentations of those structures.

3 Synchronization in LMDM

The Layered Multimedia Data Model (LMDM) [8, 9] is a comprehensive approach to modeling the multimedia compositions. The model is constructive, in the sense that it builds outward from data objects at the lowest layer to complete compositions at the highest. The second layer, the Data Manipulation Layer (DML), is concerned with specifying temporal structure, and the third, the Data Presentation Layer (DPL), is concerned with describing presentation.

Temporal alignment is handled at the DML. Multimedia data objects from the previous layer are *temporally bound* to each other using a small set of ordering operations, of which the primary two are concatenation and temporal overlay. Using data objects like building blocks, and operations as glue, temporal structures are built. LMDM allows temporal alignment specifications in this way. [8]

Specification of stream synchronization is split between the the DML and DPL. [8, 9]. Synchronization granularity is specified at the DML, although separately from temporal alignment. Thus, granularity is defined independent of presentation. A single temporal structure defined at the DML may have many different presentations, and thus many different interpretations of its synchronization constraints.

Constraints take the form of synchronization points. A synchronization point is actually a set of points at fixed indices in two or more streams. When a synchronization point is reached during the playback of one stream, the other streams which are indexed by that point are checked to see if they have yet reached that point. If not, the playback device must take some action to resynchronize the streams.

Exactly what action is to be taken is described at the DPL. There are several possible actions to choose from [9, 2], each of which may be useful depending on the playback environment. The DPL also specifies whether a tolerance will associated with a particular synchronization point. Tolerance, in this case, allows two streams to be considered synchronized as long as the trailing stream is within a certain Δ of the appropriate synchronization point, even if it has not reached it yet.

As LMDM allows a many-to-many mapping between temporal structures and presentations, not only may a set of synchronization points have different interpretations depending on the presentation, but a single set of strictness rules many be applied to a variety of temporal structures. This can be especially useful if a particular environment has consistent performance, since the strictness rules may be associated with all presentations in that environment.

4 Conclusions

From the viewpoint of authoring-oriented models, there are important conceptual separations between temporal alignment and stream synchronization, and between synchronization granularity and strictness. LMDM supports these conceptual separations, and provides an appropriate, general model for each kind of constraint.

References

[1] Arbab, F., Herman, I., and Reynolds, G. An object model for multimedia programming. In *Proceedings of Eurographics '93*, pages C101–C113, 1993.

[2] Blakowski, G., Hubel, J., and Langrehr, U. Tools for specifying and executing synchronized multimedia presentations. In *The 2nd International Workshop on Network and OS Support for Digital Audio and Video*, pages 271–282, 1991.

[3] Eun, S.B., No, E.S., Kim, H.C., Yoon, H., and Maeng, S.R. Eventor: an authoring system for interactive multimedia applications. *ACM Multimedia Systems Journal*, pages 129–140, September 1994.

[4] Gibbs, S. Composite multimedia and active objects. In *Proceedings of OOPSLA '91 Conference*, pages 97–112, 1991.

[5] Hardman, L., Bulterman, D., and van Rossum, G. The Amsterdam hypermedia model: Extending hypertext to support real multimedia. *Hypermedia Journal*, 5(1):47–69, 1993.

[6] Hoepner, P. Synchronizing the presentation of multimedia objects. *Computer Communications*, pages 557–564, November 1992.

[7] Little, T.D.C. and Ghafoor, A. Synchronization and storage models for multimedia objects. *IEEE Journal on Selected Areas in Communications*, pages 413–427, April 1990.

[8] Schloss, G. and Wynblatt, M. Building temporal structures in a layered multimedia data model. In *Proceedings of the ACM Multimedia Conference*, pages 271–278, October 1994.

[9] Schloss, G. and Wynblatt, M. Presentation layer primitives for the layered multimedia data model. In *Proceedings of the IEEE International Conference on Multimedia Computing and Systems*, May 1995.