On the Design of Multimedia Interchange Formats[1]

John F. Koegel
Interactive Media Group
Department of Computer Science
University of Massachusetts--Lowell
Lowell, MA 01854
koegel@cs.ulowell.edu

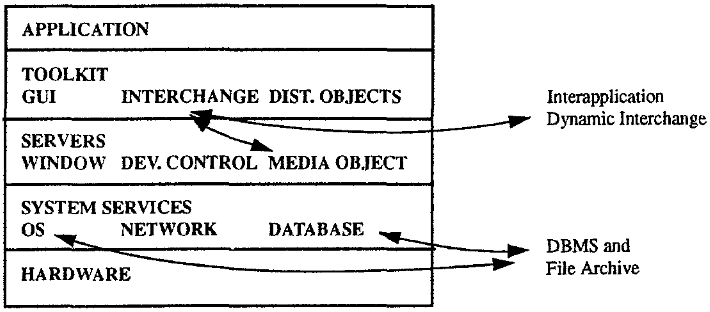
ABSTRACT

Although it is generally believed that a standard interchange format will play a crucial role in the growth of the multimedia application market, different proposals currently being developed have divergent models and each appears to have primarily evolved using an experimental methodology. In this paper we first provide a survey of these current efforts, discussing goals, architecture, and abstractions of each format. We provide a comparative summary in terms of representational capability and functionality. We classify the composition models as either track oriented or object oriented, and use this distinction to clarify differences in inter-object referencing, compositionality, and access/presentation procedures. We enumerate features that would enhance the ability of a format to support real-time interchange, and conclude with an overview of an approach to rigorously evaluate and compare such format models. This approach is based on a set of benchmark interchange cases and various parameters to be measured in a performance test.

1. Overview

In order for multimedia applications to work together and realize the benefits of distributed computing, a common interchange format for multimedia information is needed. It is not sufficient for the individual media formats to be standardized. The temporal, spatial, structural, and procedural relationships between the media components are an integral part of multimedia information and must also be represented. Today there is a growing realization that lack of a common format is a serious impediment to the development of the market for multimedia applications. Without a representation that is widely adopted and is sufficiently expressive, multimedia content that is created in one application can not be read or reused by another application. Further, in order for multimedia information to be used on several platforms, each application-defined format must have a converter on each platform.

FIGURE 1. Multimedia Interchange Context



In the architecture of multimedia systems, interchange appears in three different modes (Figure 1):

1. *Interapplication interchange:* two or more applications exchange multimedia information using either an interchange API or a distributed object API.

[1] Third International Workshop on Network and Operating System Support for Digital Audio and Video, November 12-13, 1992, La Jolla, California.

2. *Archive:* an application saves and accesses multimedia information through a file system or DBMS, also using an interchange API. Contemporary multimedia authoring packages follow this case.
3. *Presentation:* A media object server provides media objects to distributed clients in a networked environment. An example application is video-on-demand.

The design of multimedia interchange formats can also be viewed in the context of the interchange format hierarchy (Table 1). In this diagram levels correspond to increasing specificity. At the top level are general container formats. These formats are application independent. At the bottom level are media specific formats. These formats are optimized for a specific media type.

TABLE 1: The Multimedia Interchange Format Spectrum

Category	Examples
General container	GDID, ASN.1, Bento
Metalanguage	HyTime
Multimedia document architecture	HyTime DTD
Special purpose object container	QuickTime Movie File, MHEG, OMFI
Monomedia	MPEG, JPEG Script languages

The major technical issues that must be addressed include:

1. *Multimedia data model:* A data model for structured time-based interactive media (multimedia and hypermedia) including temporal composition, synchronization, multiple media formats, addressing of media objects and composite media objects, hyperlinking, and an input model for interaction.
2. *Scriptware integration:* Many authoring tools integrate multimedia data with specialized procedural scriptware which may be text based or iconic languages. These tools have a tight association of scripts with media objects and media composites, in particular associating input semantics of input objects with script input processing. The interchange formats must retain the associations of the scriptware and the media objects. Further, scripts must be able to reference structured media objects for attribute control, retrieval, and presentation.
3. *Storage efficiency:* An encoding should be efficient for storage, but the container is a small fraction of the information in a typical multimedia presentation.
4. *Access efficiency:* An encoding should be efficient for time-constrained and resource-limited retrieval. Enhanced functions for progressive and multi-resolution delivery, flexible storage organization, media interleaving, index tables, and partial media referencing can support this goal.
5. *Portability:* GUI and platform architecture independence are essential, preferably without penalizing interchange on a single platform. Issues include look and feel independence, input architecture independence, file and object referencing, byte ordering, and data type encoding.
6. *Extensibility*: It should be possible to add new media formats, new media attribute, and other container extensions

In the rest of the paper we first provide a survey of these current efforts, discussing goals, functionality, and abstractions of each format. This review includes composition, time models, hyperlinking and input handling, architecture independence, and extensibility. This part of the presentation leads to an informal comparison to be made in the following section in the form of a function checklist. This is followed by a discussion of the role of interchange formats in supporting realtime interchange. Finally, we propose a benchmark for validating formats which are designed for multimedia interchange.