Device and Data Independence for Multimedia Presentations

Richard Staehli, Jonathan Walpole, and David Maier
Department of Computer Science and Engineering
Oregon Graduate Institute of Science & Technology
Portland, Oregon, USA
{*staehli, walpole, maier*}*@cse.ogi.edu*

October 14, 1995

The hype for multimedia systems predicts access to any type of visual or aural media on the desktop. But in this networked future, will *every* type of media be accessible from *every* terminal device? Current multimedia standards do not allow content that is authored for high-bandwidth workstations to scale down for low-bandwidth applications. The problem is that application requests are commonly interpreted as requests for the highest possible quality and resource overloads are handled by ad hoc methods. We can begin to solve this problem by specifying Quality of Service (QOS) requirements based on functionality rather than on content encoding and device capabilities.

The potential of distributed multimedia computing can be achieved by offering *device independent* and *data independent* service interfaces. Device and data independence are well known principles of database system design. In multimedia systems they have the following meaning:

- Content can be presented on devices that have different resolution and bandwidth characteristics.
- The location and encoding of stored data should be transparent to the user.

Device and data independence is already supported by some content authoring standards. For example the emerging ISO MHEG standard uses of *virtual coordinates* for

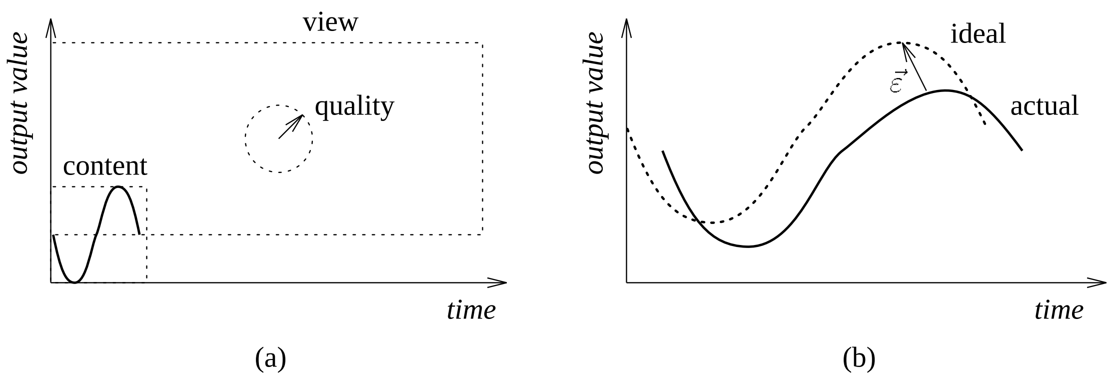

Figure 1: Content, view, and quality specify presentation QOS.

content layout [3]. However, real-time presentations typically reveal device and data dependencies through variations in presentation quality.

QOS specification. We propose a three-step methodology for QOS specifications: defining an ideal presentation, choosing an error interpretation, and constructing a user model.

An *ideal presentation* is the set of expected output values for every point in the presentation space and time. The ideal output values may vary continuously over the coordinate space of Real-numbers, unlike the actual ourput which has finite resolution and discrete values. As a consequence, the specification of an ideal presentation is device independent like a PostScript document. Figure 1(a) illustrates the specification of an ideal presentation through a *content* descriptor that may be reused in many presentations and a *view* descriptor that specifies a particular mapping of content onto device and real-time coordinates.

An *actual presentation* will deviate from the ideal because of device limitations and choice of presentation algorithms and scheduling policy. Device limitations such as screen color depth may require dithering or some other approximation of the ideal color values. Video resolution is limited by the pixel dimensions of an output window and sample rates for both video and audio are limited by device bandwidth. The choice of compression, decompression, and rendering algorithms can introduce errors in the output values. The choice of scheduling policy affects the timing of those output values. However, just as the specification of an ideal presentation is device independent, the specification of allowable error should be independent of the mechanisms used for a presentation.

An *error interpretation* maps each point in an actual presentation to a point in the ideal presentation. Figure 1(b) shows an error interpretation ε for a single point in an audio presentation. The vector $(\varepsilon_t, \varepsilon_v)$ says that the value v at time t should have occurred at time $t + \varepsilon_t$ and should have had the value $v + \varepsilon_v$. An interpretation of error in a video presentation must also account for displacement errors in both x and y.

This definition allows many different error interpretations for a given pair of ideal and actual presentations. It is tempting to define a "correct" interpretation of error based on the intended correspondence of output events with content values for a particular implementation. But we want to constrain presentation outputs, not the implementation.

Finally, presentation quality requirements can be defined in terms of a *user model.* A *user model* estimates presentation quality from an error interpretation. We have described a user model based on an error vector of *shift*, *rate*, *jitter*, and *resolution* components for each coordinate dimension, and a *synchronization* error component for the timing error between outputs [5]. These error components are defined in terms of an error interpretation. The user model estimates presentation quality for an error interpretation by normalizing each error component according to user sensitivity. The user model "accepts" a presentation if an error interpretation exists such that at no point in the presentation does the normalized magnitude of the error vector exceed a fixed threshold. This is a conservative approach. Alternative user models bound the average error or place other constraints on the distribution of error over an entire presentation.

QOS-Driven Presentations. Multimedia systems can provide better service if the QOS requirements of each client are known. A QOS specification can serve as a throttle to reduce resource use: requesting, for example, 24 frames/second video when a data source could supply 60. Resource regulation is essential in a shared environment. A QOS specification can also indicate which balance of resources provide the best quality for a particular presentation. For example, in a bandwidth constrained environment, an action video might be best presented at 320x240 pixels and 15 frames/second while a video of a chalkboard lecture uses the same bandwidth more effectively with 640x480 pixels and 4 frames/second.

Some systems will guarantee performance, others may only provide best-effort service. QOS specifications are needed to drive resource management decisions in both cases.

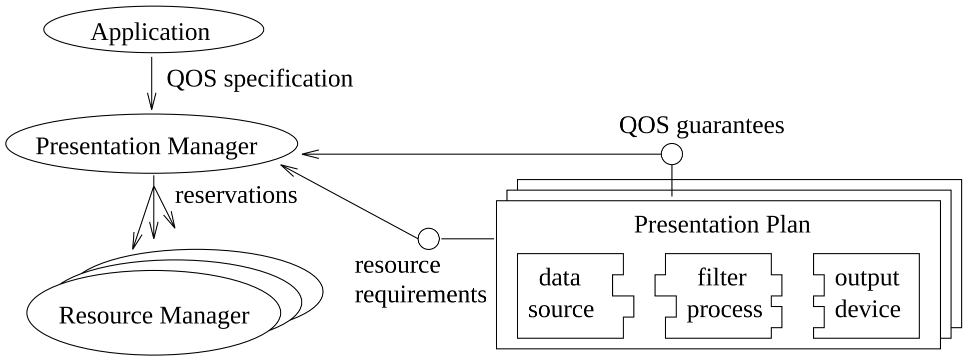

Figure 2: Presentation planning.

Best-effort resource management still involves making decisions about how to trade one kind of resource consumption for another. This is a planning problem that can be guided by the QOS specification. Making guarantees (which can be hard or statistical) requires an end-to-end resource reservation approach and an admission test [2, 1, 4]. Figuring out which of many different resource allocation plans is best is a decision that can be guided by the QOS specification.

Figure 2 illustrates a high-level architecture for an admission test. A presentation manager receives the QOS requirements for a presentation from an application. A presentation plan is *feasible* if it can guarantee the QOS requirements and if the presentation manager can reserve resources for the plan. The admission test can choose to execute the feasible presentation plan with the fewest resource requirements.

Conclusion. Multimedia systems are only beginning to realize the flexibility inherent in digital computing. More work is needed to understand QOS requirements for multimedia presentations and to exploit those requirements for optimal resource management. Device and data independent QOS specifications allow applications to say *what* multimedia services are required without restricting *how* they are implementated.

References

[1] A. Campbell, G. Coulson, F. Garcia, D. Hutchison, and H. Leopold. Integrated quality of service for multimedia communications. In *Proceedings IEEE INFOCOMM '93*, San Francisco, USA, April 1993. IEEE.

[2] M.B. Jones, P.J. Leach, R.P. Draves, and J.S. Barrera, III. Support for user-centric modular real-time resource management in the Rialto operating system. In *NOSS-DAV 95*, pages 55–66, Durham, N.H., April 1995. IEEE Communications Society.

[3] T. Meyer-Boudnik and W. Effelsberg. MHEG explained. *IEEE Multimedia*, 2(1):26–38, Spring 1995.

[4] K. Nahrstedt and J.M. Smith. The QOS Broker. *IEEE Multimedia*, 2(1):53–67, Spring 1995.

[5] R. Staehli, J. Walpole, and D. Maier. Quality of service specifications for multimedia presentations. *To appear in Multimedia Systems*, August 1995.