

Figure 17 Screen shots of an object accessed via the Smart Client (left) and via Netscape/HTTP Gateway (right)



the parent fail for some reason. The second thing that makes rapid error recovery possible is the pull-based manner in which the client extracts object data from the object servers. As the client never has a continuous open stream to an object server, and pulls in only relatively small pieces of an object for any given request, detection of a defunct object server occurs almost instantaneously. Therefore, should an object server ever fail to respond to a request for data for an object within a specified time, the client may safely assume that object server is incapable of supplying further data. Consequently, the client also makes all future requests for data for that object to an alternate object server defined in the backup-info for that object. A diagram showing the normal operation (Figure 16, top), a dropped request due to a server crash

(Figure 16, middle), and the subsequent recovery (Figure 16, bottom) for a client accessing data is shown.

The fourth and final example highlights the ability of the Media Bank to support a variety of clients with varying capabilities.

One could be forgiven for thinking that the screen shots shown in Figure 17 appear to be only standard pages of an ordinary World Wide Web browser. In reality, the browser shown on the left has been slightly modified to support the Media Bank libraries. The fact that the Mosaic browser was chosen for modification is not important; the choice made reflected the fact that it provided the most expedient means available

for generating a window for the “widgets” used by the Media Bank client libraries. The total modifications made to the Mosaic browser source code total roughly forty lines, but those forty lines provide a single point of access into the Media Bank client libraries that any smart object can use. The simplicity of this interface gives it its power, as clients can be written solely in Scheme and stored as objects that can be downloaded as needed. In effect, the Mosaic browser has been transformed from a simple display engine into a powerful virtual machine upon which any Media Bank object can be run.

Not all clients will be Media-Bank-capable, however (efforts to create a Media Bank plug-in for the Netscape browser client are ongoing at the time of writing); therefore a mechanism is needed to afford these clients access to the Media Bank. To that end a gateway application has been written that performs all of the negotiated accesses and object retrieval on the behalf of standard Hypertext Markup Language (HTML) clients. An example of the gateway operation is shown in the righthand screen shot of Figure 17. In this image, an ordinary browser is displaying the same “page” as the Media Bank enhanced Mosaic browser on the left. Unable to interpret the reference to the embedded Media Bank object in the page, this browser has been directed to an HTML gateway that acts as a proxy Media Bank client for the standard Netscape browser. The gateway itself is a client that does its best to retrieve the best possible approximation to the object requested. For example, if the object is a movie then the gateway will provide a key still from the movie, or it may instead send a small audio clip and a still. At present the backup data are supplied by the author of the object but future versions will be fully automatic and be sufficiently intelligent to tailor the approximated response according to the type of browser, and the bandwidth of the link.

Conclusions

In many respects, the Media Bank is a merger of Internet distribution techniques with television. Our goal is to find a common ground between the synchronous broadcasting of full packaged information associated with filmed entertainment and the demand-based dynamic assembly associated with computer networks. At the same time, we use the flexibility of programmable clients and servers to embed new techniques for indexing and assembling video as they arise. New applications will emerge when large populations have access to sufficiently powerful tools and

databases. The Media Bank provides the foundation upon which to build these tools and databases, as it is optimized for programmability and user control.

As a vehicle for entertainment, the bank can be the basis for a television receiver. In our demonstration systems, synchronized video images appear simultaneously within a modified World Wide Web browser window and full screen on an external television. The browser can be the program guide, the device that builds a unique program for each viewer, or simply a controller for scanning through material. Future work will enrich this interface.

For research, we are using the Media Bank to explore ways to analyze, index, and deliver high-bandwidth isochronous information such as sounds and pictures. The Media Bank model of distributed storage and pull-based delivery is a relatively new one, and as such its capabilities have yet to be fully explored. At its most basic level, multimedia is nothing more than a unified delivery channel capable of containing diverse data types. To date, by dint of its size, rigidity, and complexity, video has been excluded from the same digital pipe that has carried stills, sound, and text. Our naming structure, storage method, and access paradigm allows a diverse set of representations to be jointly cataloged, potentially simplifying the problem.

We do not envision a system such as the Media Bank replacing broadcast for entertainment in the near term. Broadcasting is both bandwidth-efficient and resonates with a great many commercial and social interests. Until our networks increase their capacity, we cannot scale the Media Bank up to the level of the audience reached by popular television shows. However, the existing model of broadcast television has never been seriously challenged by any robust, computational alternative. Further, the expanded repertoire of available programs made possible by digitally compressed delivery and the implicit presence of computing at the receiver beg the question of how the two media can work together.

**Trademark or registered trademark of DIRECTV, Inc., a unit of Hughes Electronics Corporation, X/Open Co. Ltd., or Sun Microsystems, Inc.

Cited references

1. T. J. Berners-Lee, R. Cailiau, and J. F. Groff, “The World Wide Web,” *Computer Networks and ISDN Systems* **25**, 4–5, 454–

- 459 (November 1992).
2. L. A. Rowe et al., "A Distributed Hierarchical Video-on-Demand System," *Proceedings of the 1995 International Conference on Image Processing*, Washington, DC (October 1995).
 3. L. A. Rowe, D. A. Berger, and J. E. Baldeschwieler, "The Berkeley Video-on-Demand System," *Multimedia Computing—Proceedings of the Sixth NEC Research Symposium*, T. Ishiguro, Editor, SIAM (1996).
 4. D. Berger, *Video-on-Demand Metadata Query Interfaces*, master's degree thesis, University of California Berkeley, Computer Science Division (June 1995), <ftp://roger-rabbit.cs.berkeley.edu/pub/thesis/dvberger-masters.ps>.
 5. ISO/IEC DIS 10918-1, ITU-T Rec.T.81 (JPEG), *Information Technology—Digital Compression and Coding of Continuous-Tone Still Images* (1992).
 6. ISO/IEC 11172, *Information Technology—Coding of Moving Picture and Associated Audio for Digital Storage Media as up to about 1.5 Mbit/s* (1993).
 7. ISO/IEC CD 13818-2, Recommendation H.262, *Generic Coding of Moving Pictures and Associated Audio*, Committee Draft, Seoul (November 1993).
 8. R. Jain and A. Hampapur, "Metadata in Video Databases," special issue on *Metadata for Digital Media*, ACM SIGMOD (1994).
 9. K. Dienes, *Information Architectures for Personalized Multimedia*, master's degree thesis, MIT Media Laboratory, Cambridge, MA (June 1995).
 10. W. Clinger et al., *Revised Report on the Algorithmic Language Scheme*, MIT AI Lab Memo 848b, MIT, Cambridge, MA (November 1992).
 11. N. Abramson, *Context-Sensitive Multimedia*, master's degree thesis, MIT Media Laboratory, Cambridge, MA (June 1993).
 12. A. B. Lippman, R. G. Kermode, H. N. Holtzman, and K. Dienes, *Media Bank—Communications Protocol Definition*, Internal Document, MIT Media Laboratory, Entertainment and Information Systems Group, Cambridge, MA (October 1995).
 13. K. Sollins, *Plan for Internet Directory Services*, rfc1107 (July 1, 1989).
 14. T. Berners-Lee, *Universal Resource Identifiers in WWW: A Unifying Syntax for the Expression of Names and Addresses of Objects on the Network as Used in the World-Wide Web*, rfc1630 (June 9, 1994).
 15. D. Clark and D. Tennenhouse, "Architectural Considerations for a New Generation of Protocols," *Proceedings of the ACM SIGCOMM '90* (September 1990), pp. 201–208.
 16. L. Zhang, R. Braden, D. Estrin, S. Herzog, and S. Jamin, *Resource ReSerVation Protocol (RSVP)—Version 1 Functional Specification*, <ftp://www.isi.edu/internet-drafts/draft-ietf-rsvp-spec-08.txt>.
 17. S. Deering and R. Hinden, *Internet Protocol, Version 6 (IPv6) Specification*, Internet Draft, <ftp://ds.internic.net/internet-drafts/draft-ietf-ipngwg-ipv6-spec-02.txt> (June 1995).
 18. W. R. Neuman, A. Cigler, S. M. Schneider, S. O'Donnell, and M. Reynolds, *The Television Sound Study*, MIT Media Laboratory, Cambridge, MA (1987).

Accepted for publication April 4, 1996.

Andrew Lippman MIT Media Laboratory, 20 Ames Street, Cambridge, Massachusetts 02139-4307 (electronic mail: lip@media.mit.edu). Dr. Lippman received both his B.S. and M.S. degrees in electrical engineering from the Massachusetts Institute of Technology. In 1995 he completed his Ph.D. studies at the École Polytechnique Fédérale de Lausanne, Switzerland. He is currently

associate director of the MIT Media Laboratory and a lecturer at MIT. He holds seven patents in television and digital image processing. His current research interests are in the design of flexible, interactive digital television infrastructure.

Roger Kermode MIT Media Laboratory, 20 Ames Street, Cambridge, Massachusetts 02139-4307 (electronic mail: woja@media.mit.edu). Mr. Kermode is a doctoral student at the MIT Media Lab where he earned a master of science degree in 1994. He also holds undergraduate degrees in electrical engineering and computer science from the University of Melbourne. After completing his undergraduate studies in 1990 he worked at the Telstra (Australia) Research Laboratories as a research engineer investigating cell loss recover techniques for packetized digital video. Since commencing his studies in the United States he has interned at Silicon Graphics Incorporated. He is a Fulbright Scholar and has received fellowships from AT&T and Motorola, Inc. His research interests include video coding, object-oriented representations for multimedia, computational media, and interactive network infrastructure.

Reprint Order No. G321-5606.