Situating MMDD with respect to WWW, Portfolio, PREMO, Hyper-G, ScriptX

Sha Xin Wei*

November 8, 1994

Abstract

We compare the MMDD distributed media system with World Wide Web, Portfolio, Hyper-G as well as the standards SGML, HyTime, PREMO. We also discuss MMDD's relation to the emerging class of scripting environments designed for highly interactive media-rich environments such as ScriptX.

In this note, we assume that the reader is familiar with the design principles underlying the MMDD. (See [9].)

*ASD, Stanford University, *xinwei@jessica.stanford.edu*

1 Comparison with World Wide Web and NCSA Mosaic

While we have exported some of our corpora also via World Wide Web [2], [1], WWW and NCSA's design significantly differs from the MMDD's (see section 4), and do not serve the MMDD's spectrum of media and interactivity. We briefly contrast the two designs to more clearly situate the MMDD:[1]

- Display Metaphor: NCSA Mosaic, and HTML, are based on a word-processor document model: text with inlined media "raisins," which is too rigid for the sort of interactions the MMDD is designed to support, such as staged video sprites and dynamically defined compound objects like "classroom activity" or "historical actor." The MMDD supports a world of co-equal media entities, which may be anything ranging from QuickTime movies, Hypercard stacks or Mathematica programs to streams of music or video.

- Purpose: WWW and Portfolio were designed to deliver word-processor-like *documents,* while MMDD is a toolkit designed to provide structures to inform and enrich models or *simulations* of social systems, which is a much more live, dynamic set of applications. Imagine trying implement SimCity using MS Word. WWW's hypertext model is the same as MS Word, where the documents happen to span whole continents; this is merely a difference in degree, not in quality. It would be just as inefficient to try to implement a simulation in WWW.

- Database Services: Mosaic and WWW have no native design for mediating database services. CGI's represent a relatively limited stopgap measure via static data forms which do not streamline the real work of cross-database communication, query-generation, model management, all of which are uniformly and dynamically handled by the NS dbkit's object-oriented methods. The MMDD provides an uniform programmatic interface so that all clients may invoke a single language for

[1]While we focus most of our comparison at the level of the user interfaces, it is important of course, to distinguish the underlying World Wide Web architecture from the interfaces that happen to be written for it.

multimedia search, annotation and other database functions for a large variety of database engines.

- Format. Mosaic requires that its documents be marked up in a fixed format – HTML – derived from an older version of SGML. The MMDD, via its object-oriented database structures, requires essentially no assumption about the internal structure of a blob, yet maintains meta-information and supports database functions on the blobs. The data should be exchanged in the most meaningful terms, eg. an architectural building plan with associated information, rather than at the lowest common denominator like a dead bitmap.

  Specifically, whereas links must be explicitly written into HTML documents, a laborious and error-prone operation which requires that the authors' documents be marked up, MMDD's link-maps are transparently written into independent databases by simple user gestures, such as *drag-drop* under the NS Workspace and Mac front ends.

- Interface. Mosaic front ends have frozen layout and functionality. A fundamental difference is that WWW front ends necessarily have fixed layout – whereas MMDD front ends' elements (buttons, display fields, backgrounds...), may be moved, reshaped, even redefined at any time.

  Whereas XMosaic and Windows require installation of viewers which must conform to some Mosaic-specific protocols, MMDD front ends exchange structured data with ordinary commercial applications familiar to users.

- Abstractions or meta-data representation. WWW has no notion at all of defining custom schema or schema editing. By contrast, the MMDD is designed to let authors revise their own schema. MMDD interfaces can be reconfigured in minutes.

- Services. In contrast to WWW's strategy of providing libraries including special-case media viewers on an *ad hoc* basis, the MMDD's object architecture provides a rational and flexible framework (described below) for integrating any service, not just format conversion, but arbitrary functions such as performing some calculus or doing a fractal decomposition of a videoframe.

- Generality. Finally, WWW may be viewed as a lowest common denominator document delivery service which fits inside the MMDD architecture. In fact, the MMDD uses a WWW server to its archived media to WWW browsers. Any MMDD project's media can automatically be made available through WWW. In this context one must realize that merely dumping a file into a WWW server is the relatively easy aspect of multimedia programming. It takes considerable manual effort to produce hypermedia, especially in a static data structure like HTML. And even more to create useful indices, search engines, models and simulations atop the media. To develop interactive multimedia "kiosks" with such labor-intensive means mis-allocates human labor to tasks which can be automated in a system like the MMDD. Example tasks include automatic filtering, automatic link generation from MMDD search objects, than explicitly manually pre-fabricating links.

2 Comparison with Portfolio

Some of the same differences are true for Portfolio, too.

- The MMDD deliberately deferred issues of user authorization and versioning to Portfolio. Unfortunately Portfolio in its original version was designed around gopher-like file-based description which is too limited for any sort of flexible descriptive systems. Portfolio is superceded by WWW.

- Another difference is that Portfolio's databases are not designed to communicate with external clients, so it cannot co-exist with infromation retrieval applications written by non-Portfolio programmers. This may be resolved in the future as they develop and stabilize an API.

- A third, fundamental difference is that Portfolio front ends are designed specifically for the Macintosh OS, so there is no way for anyone using other operating systems to access the information. This is now being supplanted by WWW browsers, but see remarks above.

3 Comparison with PREMO, HyTime, SGML standards

PREMO[8], HyTime [4], SGML [3] are standards, not working systems, but we include them because they represent current standards in the formal description of multimedia (including time-based media) data. [2] The MMDD does not prefer one format to another, and in fact is predicated on the existence of a rich space of transformations between formats. We discuss these standards from the point of view of how well they serve to describe the kinds of interactive, media-rich and structure-rich environments we wish to support.

3.1 SGML

SGML probably is the richest description of structured text. Principal limitations of SGML include

- It is designed for one-dimensional character streams. Even multi-linear muscial scores would be difficult to describe.
- SGML in its original form does not deal with time-based media, or other non-text media (3D animation, mathematical expressions),
- SGML allows no procedural descriptions; it is a meta-level description of data formats rather than a full language.
- SGML has no abstractions for user interaction.

3.2 HyTime

HyTime adds descriptions of time-based media to SGML, and includes notions of synchronization, a important problem in time-based media. However, as pointed out in [?], there were no procedural scripting facilities, nor any models of user interaction.

[2]PREMO is the only ISO standard for presentation systems.

3.3 PREMO

We feel that the MMDD would interpret the PREMO scheme because it is already layered to decouple data, model and presentation [9]. As mentioned in [HyTime], however, it is not clear how well HyTime or PREMO match the expressivity of applications like Apple Hypercard, Oracle Media Objects, or emerging frameworks like Kaleida's ScriptX or Taligent's time-based media.

4 Comparison with Hyper-G

Hyper-G [5] at the University of Graz, Austria, shares design tenets with our work. They also decouple data, format, and hypermedia link structure. In fact they have a link server which seems to be more powerful than our link database. Significant differences are

- Hyper-G is built around a traditional metaphor of documents with embedded links.
- It is essentially a unix/X windows system and thus, does not integrate with commercial universe of personal computer applications. As such, Hyper-G deals with a set of environments and users complementary to those served by MMDD.
- Hyper-G has no scripting framework of GUI's, whereas MMDD front ends use Hypertalk – a popular scripting language and AppleScript on the Macintosh, and InterfaceBuilder/ProjectBuilder on NS.

5 Comparison with interface scripting frameworks

The MMDD was designed to support the creation of highly interactive media-rich simulations. Currently the industry standard remains large monolithic applications such as Hypercard and Supercard and Director. When scripting environments are developed which are dissolved into the substrate operating system, distributed media systems like the MMDD will come into their own.

5.1 ScriptX

Kaleida Labs' ScriptX [6] represents probably the state of the art in animation frameworks, and would be a natural extension for the MMDD's current front end kits. It has an object animation system which works across all platforms supporting the runtime environment (first Macintosh and IBM PC), and an extremely powerful scripting language.

At this time, however, ScriptX is a standalone system designed for single hosts. Kaleida intends to open a back door to external media-bases, and produce a distributed objects version of ScriptX in subsequent releases. At that point, we will consider a direct connection to ScriptX.

5.2 TCL/TK, others

Other advanced interaction scripting systems include TCL/TK [7], AthenaMUSE (MIT CECI) and perhaps General Magic's Telescript. TCL/TK's main platform is X Windows. Again system issues such as lack of global user interface design (grab bag of GUI "widgets"), and lack of object-interoperation tend to undercut the power of TCL applications. However it is interesting to note that TCL/TK has been bound weakly to some NeXTSTEP and Mac development environments. Most of the other scripting langauges such as PERL, Python, Dylan, SK8 are restricted to particular OS's. In particular we seek frameworks which span Macintosh and unix environments, yet can extend to farily arbitrary future OOPE's.

We will investigate other scripting environments as they mature.

References

[1] T. J. Berners-Lee. The world wide web, talk at Stanford PCD seminar CS 547, 1993.

[2] T. J. Berners-Lee, R. Cailliau, J-F Groff, and B. Pollermann. World-wide web: The information universe. In *Electronic Networking: Research, Applications and Policy*, volume 2, pages 52–58, 1992.

[3] Charles F. Goldfarb. *The SGML Handbook*. Oxford University Press, 1991.

[4] ISO. Hypermedia/time-based document structuring language (hytime). Technical report, ISO/IEC, April 1992.

[5] Frank Kappe, Keith Andrews, Joerg Faschingbauer, Mansuet Gaisbauer, Michael Pichler, and Juergen Schipflinger. Hyper-g: A new tool for distrobuted hypermedia. Technical report, Institute for Information Processing and Computer Supported New Media (IICM), Graz University of Technology, Graz Austria, 1994.

[6] Kaleida Labs. Kaleida scriptx white paper. Technical report, Kaleida Labs, 1994.

[7] J. K. Ousterhout. An X11 toolkit based on the TCL language. In *Proceedings of the Winter 1991 USENIX Conference*, pages 105–115, 1991.

[8] PREMO. Premo: An iso standard for multimedia presentations. In *Proceedings ACM Multimedia Conference*, 1994.

[9] Sha Xin Wei. An open testbed for distributed multimedia databases. In *Proceedings AAAI Workshop on Indexing and Reuse in Multimedia Systems*, 1994.