Standardizing Hypertext: Where Next for OHP?

David Millard, Hugh Davis and Luc Moreau

Intelligence, Agents, Multimedia,
University of Southampton, Dept. of Electronics and Computer Science,
Southampton, U. K.

Abstract. Over the last six years the Open Hypermedia Systems Working Group (OHSWG) has been working in a coordinated effort to produce a protocol which will allow components of an Open Hypermedia System to talk to one another in a standardised manner. In this paper we reflect on this work and the knowledge that has come out of it, evaluating the differant approaches to standardisation in the light of our experiences. We discuss the problems we encountered and redefine the goals of the effort to be more realistic, presenting the Fundamental Open Hypermedia Model (FOHM) as an example of this more realistic approach. Finally we describe a possible future path that encompasses the research interests of the OHSWG while still leading ultimately to interoperability.

1 History of the OHP Effort

1.1 Original Proposal

The First Workshop on Open Hypermedia [25] was held at Edinburgh in conjunction with ECHT'94. This workshop was concerned with the growing class of hypermedia systems such as Chimera [2], DHM [9], HyperForm [24], Microcosm [5], Multicard [20] and the HB/SP series [21], which clearly separated hypertext structure (links) from the content (documents). The participants in this workshop were keen to provide hypertext link services which could provide hypertext structure for documents which were displayed using existing desktop applications such as Word for Windows and Emacs. This workshop lead to the formation of the Open Hypermedia Systems Working Group (OHSWG), the full history and rationale behind the work of this group can be viewed on their web pages [1].

An interesting finding of this first workshop was that although the major area of interest for the participating research groups was the design and implementation of link servers, most of their time was being spent on the implementation of clients; the researchers were spending significant effort producing text and graphics clients for the link services, either by writing them from scratch or writing macros to adapt existing desktop applications. A proposal from Antoine Rizk was that the group could contribute by producing a lightweight message based protocol that could be used to communicate about simple link service functions. The rationale was that all link services had an approximately similar data model and that the operations that the link services could perform were also similar; all that would be required was a simple "shim" (protocol converter) that could convert between the client protocol and the server protocol, and then it would be possible for groups to share client implementations. The idea was simple and lead to the production of the first draft of the Open Hypermedia Protocol (OHP) [6] which was presented at the next workshop in 1996.

What happened next to OHP may well be a familiar story to other groups who have attempted to produce an application level protocol. The committee effect started to take hold and the protocol grew; there were discussions about whether we were going to use a message passing interface or an API; there were arguments about the on-the-wire protocol to be used, and the group became confused about aspects of resource location and naming instead of concentrating on hypertext. Furthermore, the increasing influence of the World Wide Web throughout this period tended to change original assumptions by producing a system that was open in very different ways from the OHSWG systems.

However, there were some good outcomes from this stage of the work. The scope of the project changed from attempting to provide a lightweight communication mechanism for shared clients and heterogeneous link servers, to attempting to create a reference model and implementation for open hypertext systems. A standardized data model and basic set of operations was agreed and the groups concerned produced native OHP link servers, and agreed a temporary on the wire protocol that made possible a significant demonstration at Hypertext '98, and a paper on the experiences to that stage [18].

1.2 The Hypertext 98 Demonstration

Two systems were developed for a demonstration of interoperability at Hypertext 98 (held in Pittsburgh, USA). One from the University of Arhus, Denmark, and the other developed at the Multimedia Research Group (MMRG) at the University of Southampton.

During the development of these systems several problems became evident. Both with the protocol itself and also, more importantly, with the scope of the original draft proposal. The original draft was meant as a standard interface between clients and servers, to allow software reuse. This has increased in scope dramatically and become an all-encompassing effort to understand the nature of hypermedia and thus produce standards to provide for it. However it was soon understood that such a large goal was impossible to realise within a single protocol and as a result the protocol was split into several domains, each domain dealing with a particular type of hypermedia.

The original OHP protocol was therefore renamed OHP-Navigational (OHP-Nav) and reduced in scope to deal exclusively with navigational (node/link) hypermedia. Other domains were envisaged such as Spatial Hypermedia [12] (OHP-Space) and Taxonomic Hypermedia [22] (OHP-Tax).

The protocol itself had been originally based on the Microcosm message format [8], a sequence of tag/value pairs. However this proved difficult to parse so the OHSWG adopted XML as a suitable format [3] and the OHP-Nav message set and hypermedia objects were all defined as XML elements in a Document Type Definition (DTD).

1.3 The Hypertext 99 Demonstration

At the OHSWGs meeting at Southampton (OHS 4.5) it was decided that as the Hypertext 98 demonstration had formed such a positive focus point for the group a similar demonstration should be attempted for Hypertext 99 in Darmstadt, Germany. It was also decided that since we had demonstrated interoperability at Hypertext 98 we should now concentrate on showing some of the features of the protocol, ironically removing the need to interoperate.

Some of the more successful parts of the Hypertext 98 demonstration were the collaboration aspects. So the Danish contribution to the Hypertext 99 demonstration was an extension of this simple support into a more advanced system called 'Construct'. The Southampton contribution was to investigate a definition of computational links, where opaque computation objects are included in the hypertext model and can be referenced similarly to other objects. This resulted in a component based system known as 'Solent'.

Another system demonstrated at Hypertext '99 was CAOS [19], a spatial hypermedia collaborative system. Discussions within the working group turned to the definition of OHP-Space, starting us thinking about whether the different domains were actually that different after all.

2 What were the Problems with OHP?

2.1 What were we Trying to Standardize?

It has already been mentioned that the purpose of OHP has changed a great deal since the protocol first appeared. Initially a basic client-server communication protocol it has grown to reflect the concerns of a large community of researchers. Even given that we understood what functionality we were going to standardize there is still the question of what actually will become standard in the system. I.e. how will components actually talk to one another? There are two approaches:

1. *A programming API* : By using a standardised API client source code compatibility is preserved, whatever the servers involved, and server source code remains valid whatever the clients. This requires specifying:
   (a) the system calls,
   (b) the callbacks to be used,
   (c) the data to be exchanged.

   Examples would be to use CORBA and its interface definition language IDL, Microsoft DCOM or a Java component system using Java Beans.
   One should note that in this approach the data representation is dependent on the binding (i.e. the IDL compiler for the chosen language and the chosen implementation of the communication module).

2. *An on-the-wire communication model* : This involves defining:
    (a) the syntax of the messages (e.g. an XML hierarchy),
    (b) a set of requests, associated responses and their syntax,
    (c) the data and its syntax,
    (d) how to setup the transport medium (e.g. opening socket on a port, etc.).

At one time or another both approaches have been argued for. However the need to produce communicating systems that actually worked resulted in the group returning to the on-the-wire approach, even though the API approach seems cleaner and allows us to concentrate on the hypertext issues rather than the networking ones. The API approach also preserves source code compatibility. The implementers just need to implement two APIs, one for the client and another for the server. If they then find that a new on-the-wire protocol should be used, they can change their implementation without altering the source code of the components involved.

However it is still not a perfect solution as it may require recompiling applications when a different medium is required. This is indeed the case with CORBA where binary applications are ORB dependent. This does not give a lot of freedom to the final user as a binary is typically compiled for a fixed communication medium.

Recently, Southampton researchers have been experimenting with implementing hypertext functionality on top of agent frameworks; in particular, the Java-based SoFAR [15], the Southampton Framework for Agent Research, was the focus of this experiment. SoFAR adopts an abstract communication model, where agents communicate using "virtual channels", identified by a startpoint and an endpoint; the latter is a client-side proxy used to initiate communications and the former is a server-side entity, extracting messages from the communication channel and passing them on to agents. Startpoints are specified by an interface, and agents communicate by activating methods of this interface. Every communication module provides its own implementation of the startpoint and endpoint interfaces, relying on a specific communication mechanism (rmi, encrypted communications, . . .).

This modular organisation of the system preserves binary code compatibility of applications. Indeed, applications do not have to be recompiled when new communication modules are introduced. When an application starts a communication with a startpoint, for which the code is not loaded in memory, SoFAR and the JVM are able to load the required implementation dynamically. This property requires the language and/or the operating system to be able to load binary code dynamically. In the absence of such a facility, SoFAR would revert to a source code compatibility: it would require applications to be recompiled for every new communication module, in a similar fashion to the CORBA/IDL model.

As the on-the-wire communication model is typically adopted for Internet protocols (normally ASCII and socket based), there is a simple argument that says that since it works for the World Wide Web and the Internet it can work for OHP to! Unfortunately this approach has several disadvantages:

1. Writing efficient socket communications is a very difficult task, involving threads, polling, etc. It is very easy to produce inefficient communication systems.
2. Such libraries have to be rewritten for every application. This results in the risk of a bad implementation, where data is not properly formatted or parsed. The CORBA approach with a stub compiler avoids this problem by generating code automatically.
3. It becomes extremely difficult to deal with non-protocol data and requests, such as routing information for mobile agents, garbage collection or session management.

The DLS approach [4] is an instance of the on-the-wire protocol standardisation, except that the DLS does not specify but reuses off-the-shelve protocols or communications medium, such as sockets, http, XML, LDAP, SoFAR or tuple space.

Both approaches to interoperability have their advantages and disadvantages. On the wire protocol has the "taste" of the Internet community and a simplicity that is very appealing, while the programming API allows further techniques to be transparently added (mobility, etc.). In both cases, a data model has to be adopted. The data model specifies the type of data and its associated meaning (in terms of protocol primitives) exchanged during communications. The data model does not specify the syntax of data (this depends on the approach: ontologies in SoFar, or XML over sockets).

Remark. The data model does not force components to adopt such a representation internally: it simply requires them to *exchange* such data. Once defined the data primitives can serve as a guide to specify requests.

Without precluding any approach, API or on-the-wire, is the need to specify a powerful data model that supports all of the hypertext features that need to be standardised.

2.2 A Communications Infrastructure

Even given a data model and some communication medium, there remains the need for some type of infrastructure over which that model can be discussed by a variety of components. This infrastructure is different from the network and itself may run independently over different lower network protocols (sockets, rmi, etc.). In effect it is a framework in which components can discover each other and exchange data.

In OHP this is represented by the message headers and bodies (although not the requests themselves). Using this basic framework, OHP allows a component to send multiple requests in a single message and track messages around the system. It was also reasoned that in any complex system total semantic understanding by all components was unobtainable. To avoid the problems this would cause it was at one time argued that performatives [7] [11] should be added to the OHP header. Performatives describe the overall intention of message in concise clear terms that all components of the system can understand (e.g. this is an information message or this is a request). In this way if the component does not understand the content of the message it can still reason about the intention and make decisions about the message as a result (even if that is just to forward the message somewhere else).

In an effort to keep the OHP definitions as simple as possible performatives were never formally added. However we have never lost the belief that they are useful. In fact we now believe that disagreements within the group regarding performatives were a symptom of a much larger and fundamental problem. In the OHP specification no clean separation was made between the communication infrastructure and the definition of the hypermedia model and its operations. As a result we failed to notice that many of the problems we faced were not ours to solve in the first place!

2.3 Is OHP too Large in Scope?

One of the noticeable things about the whole OHP effort is the way that the scope of the original proposal has increased out of all proportion compared to its original intent. The OHP has grown from a simple protocol that would allow standardized clients to talk to any OHS into a mammoth undertaking that involves all of the components of a system and which includes multiple domains of hypertext and many levels of functionality (e.g. computations and collaboration). Much effort has since been directed into breaking this huge problem domain into manageable chunks, creating OHP-Nav, OHP-Space, OHP-Tax and a whole host of associated protocols. The problems we have faced would seem to indicate that the scope of the protocol should be dramatically reduced.

2.4 Is OHP too Small in Scope?

Considering the size of the task now before us this may seem like a ridiculous question. However the goals of the protocol have been moving since its inception and perhaps it is time we re-examined exactly what they are. When OHP was conceived the OHS architecture considered was a client/server one, ideal for intra-LAN systems. As technology has moved on, and we move into an age of distributed information and intelligent agents, it is possible that by increasing that scope we actually 'offload' some of the bigger difficulties to the places where they belong. It is not the job of the OHSWG to create distributed computing environments, any more than it is to create a language for exchanging knowledge.

In other words, by accepting that the scope of OHP is actually the massively distributed management and navigation of knowledge we no longer have to deal with any of the communication infrastructure issues mentioned above. Instead we have to build OHP on top of existing networks and prototype frameworks that support the dynamic exchange of knowledge between distributed components.

In effect OHP would become a semantic language that could be implemented via a variety of syntactical languages over existing communication infrastructures.

3 Where does FOHM fit in?

The Fundamental Open Hypermedia Model (FOHM) [13] is an abstract data model that supports arbitary associational information and can be used to represent Spatial, Navigational and potentially Taxonomic structure. The relationship between these domains is defined formally and interoperability is achieved by mapping from one domain to FOHM and then back to a different domain.

3.1 Where does FOHM come from?

One of the components in the Solent system used for the second demonstration at Hypertext '99 was one that allowed the storage of arbitrary XML [16]. This component was very versatile and totally reusable as it understands nothing of the data structures it stored, only the structure of the XML itself, effectively an element tree. The problem with this approach was that the storage component was extremely slow as a result, taking a much longer time to retrieve a structure via pattern matching than a database would with structures it really understood (and had presumably indexed).

As a result of this experience, when we noticed that there was a great deal of overlap between the various domains of hypertext we decided to find the highest level of structure that worked across them all. Each domain could then be represented in that structure and cross-domain interoperability would be achieved.

3.2 What is FOHM in relation to OHP?

FOHM is an attempt to concentrate firmly on hypertext data structures and is based on the OHP-Nav data model, although it inherits none of the OHP protocol definition itself (headers, application requests, etc.). It is at this data model level that we believe the efforts of the OHSWG should be focused in the future.

Although we do not suggest that FOHM replaces the OHP model we do think that OHP could benefit from the lessons that FOHM teaches. That a powerful generic structure that models all the major domains is not only possible, but it results in a versatile hypertext environment that is greater than any single domain.

4 Where next for OHP?

Before we can move forward with OHP we have to acknowledge the success we have already had with the protocol. It has managed to get a wide group of researchers to discuss the technology of hypertext on a common level and forms a basis for discussion only rivaled by the Dexter model [10]. We now have a well defined model for Navigational hypertext and are beginning to understand how this relates to other domains, such as Spatial and Taxonomic Hypertext. We also have a much greater understanding of the way our different systems are built and crucially can begin to understand the infrastructure that we require before true interoperability is possible.

As a result of these efforts the OHSWG is now in the position to finalise the existing OHP draft. This would provide a milestone by which other efforts could be judged, but we believe that it would still be a mistake. One which risks knocking OHP into obsolescence. Instead the question is how can we begin to incrementally move the effort forward again using the data model as a base?

In this paper we have argued that the most successful part of OHP has been the definition of an abstract data model and have explained how FOHM extends this model formally, providing a core data model and set of operations. Although this core can already be implemented by binding it to an appropriate infrastructure, it still needs to be extended to include the notion of perspective from Taxonomic hypertext as well as investigating other concerns, such as the interaction of the model with existing multi-user and security systems

It would be useful to produce several example bindings for different infrastructures. At a basic level the core data model makes this a trivial task, but that doesn't mean that all implementations are trivial! There is still room for interesting work on the distribution and architecture of implementations and the consequences that they might have for the model. Also there are other implementation concerns that remain beyond the scope of the model, for example the problems associated with naming [17].

Finally it is entirely appropriate to build other structures based on the core to deal with other hypertext issues, such as the provision for collaboration [23] and computation [14] within the system.

We may be building hypermedia systems but the navigation of hyperspace is much more then point and click, it involves a true understanding of how information spaces work, how they can be represented in different ways and how individual hyperwebs can be manipulated in a global information network. Above all else the goal of the OHP effort should be to increase that understanding.

References

1. Open Hypermedia Systems Working Group (OHSWG) home page. http://www.ohswg.org.
2. Anderson, K. M., Taylor, R. N., and Whitehead, E. J. Chimera: Hypertext for heterogeneous software environments. In *ECHT '94. Proceedings of the ACM European conference on Hypermedia technology, Sept. 18-23, 1994, Edinburgh, Scotland, UK* (1994), pp. 94–197.
3. Bray, T., Paoli, J., and Sperberg-McQueen, C. M. Extensible markup language (XML). Tech. rep., World-wide Web Consortium (W3C) Recommendation, Feb. 1998.

4. Carr, L. A., De Roure, D. C., Hall, W., and Hill, G. J. The distributed link service: A tool for publishers, authors and readers. *World Wide Web Journal 1*, 1 (1995), 647–656.
5. Davis, H. C., Knight, S., and Hall, W. Light hypermedia link services: A study of third party application integration. In *ECHT '94. Proceedings of the ACM European conference on Hypermedia technology, Sept. 18-23, 1994, Edinburgh, Scotland, UK* (1994), pp. 41–50.
6. Davis, H. C., Rizk, A., and Lewis, A. J. OHP: A draft proposal for a standard open hypermedia protocol. In *Proceedings of the 2nd Workshop on Open Hypermedia Systems, ACM Hypertext '96, Washington, D.C., March 16-20. Available as Report No. ICS-TR-96-10 from the Dept. of Information and Computer Science, University of California, Irvine* (1996), U. K. Wiil and S. Demeyer, Eds., pp. 27–53.
7. FIPA. Fipa 97 specification, part 2: Agent communication language. Tech. rep., Foundation for Intelligent Physical Agents, Geneva, Switzerland, Nov. 1997.
8. Fountain, A. M., Hall, W., Heath, I., and Davis, H. C. MICROCOSM: An Open Model for Hypermedia With Dynamic Linking. In *Hypertext: Concepts, Systems and Applications (Proceedings of ECHT'90)* (1990), A. Rizk, N. Streitz, and J. André, Eds., Cambridge University Press, pp. 298–311.
9. Grønbæk, K., and Trigg, R. H. Design issues for a dexter-based hypermedia system. *Communications of the ACM 37*, 3 (Feb. 1994), 40–49.
10. Halasz, F., and Schwartz, M. The dexter hypertext reference model. *Communications of the ACM 37*, 2 (1994), 30–39.
11. Labrou, Y., and Finin, T. A proposal for a new KQML specification. Tech. Rep. TR CS-97-03, Computer Science and Electrical Engineering Department, University of Maryland Baltimore County, Baltimore, MD 21250, Feb. 1997.
12. Marshall, C. C., and Shipman, F. M. Spatial hypertext: Designing for change. *Communications of the ACM 38* (1995), 88–97.
13. Millard, D. E., Moreau, L., Davis, H. C., and Reich, S. FOHM: A fundamental open hypertext model for investigating interoperability between hypertext domains. In *Proceedings of the '00 ACM Conference on Hypertext, May 30 - June 3, San Antio, TX* (June 2000).
14. Millard, D. E., Reich, S., and Davis, H. C. Dynamic service discovery and invocation in OHP. In *Proceedings of the 5th Workshop on Open Hypermedia Systems, ACM Hypertext '99 Conference, Darmstadt, Germany, February 21-25. Available as Report No. CS-99-01 from the Dept. of Computer Science, 6700 Aalborg University Esbjerg, Denmark* (1999), U. K. Wiil, Ed., pp. 38–42.
15. Moreau, L., Gibbins, N., DeRoure, D., El-Beltagy, S., Hall, W., Hughes, G., Joyce, D., Kim, S., Michaelides, D., Millard, D., Reich, S., Tansley, R., and Weal, M. SoFAR with DIM agents. An agent framework for distributed information management. In *The Fifth International Conference and Exhibition on The Practical Application of Intelligent Agents and Multi-Agents, April 10 - 12, 2000, Manchester, UK* (Apr. 2000).
16. Reich, S., Griffiths, J. P., Millard, D. E., and Davis, H. C. Solent — a platform for distributed open hypermedia applications. In *Database and Expert Systems Applications. 10th Intl. Conference, DEXA 99, Florence, Italy* (Berlin/Heidelberg/New York, Aug. 1999), T. Bench-Capon, G. Soda, and A. M. Tjoa, Eds., vol. 1677 of *LNCS*, Springer, pp. 802–811.
17. Reich, S., Millard, D. E., and Davis, H. C. Naming in OHP. In *Proceedings of the 5th Workshop on Open Hypermedia Systems, ACM Hypertext '99 Conference,*

Darmstadt, Germany, February 21-25. Available as Report No. CS-99-01 from the Dept. of Computer Science, 6700 Aalborg University Esbjerg, Denmark (1999), U. K. Wiil, Ed., pp. 43–47.

18. Reich, S., Wiil, U. K., Nürnberg, P. J., Davis, H. C., Grønbæk, K., Anderson, K. M., Millard, D. E., and Haake, J. M. Addressing interoperability in open hypermedia: The design of the open hypermedia protocol. *New Review of Hypermedia and Multimedia* (1999). Accepted for publication.
19. Reinert, O., Bucka-Lassen, D., Pedersen, C. A., and Nürnberg, P. J. CAOS: A collaborative and open spatial structure service component with incremental spatial parsing. In *Proceedings of the '99 ACM Conference on Hypertext, February 21-25, 1999, Darmstadt, Germany* (Feb. 1999), pp. 49–50.
20. Rizk, A., and Sauter, L. Multicard: An open hypermedia system. In *ECHT '92. Proceedings of the ACM conference on Hypertext, November 30-December 4, 1992, Milan, Italy* (1992), pp. 4–10.
21. Schnase, J. L., Leggett, J. L., Hicks, D. L., Nuernberg, P. J., and Sánchez, J. A. Open architectures for integrated, hypermedia-based information systems. In *HICSS 94 — 37th Annual Hawaii International Conference on System Science.* (1994).
22. van Dyke Parunak, H. Don't link me in: Set-based hypermedia for taxonomic reasoning. In *Proceedings of the '91 ACM Conference on Hypertext, Dec. 15-18, 1991, San Antonio, TX* (1991), pp. 233–242.
23. Wang, W., and Haake, J. M. Implementation issues on ohs-based workflow services. In *Proceedings of the 5th Workshop on Open Hypermedia Systems, ACM Hypertext '99 Conference, Darmstadt, Germany, February 21-25. Available as Report No. CS-99-01 from the Dept. of Computer Science, 6700 Aalborg University Esbjerg, Denmark* (1999), pp. 52–56.
24. Wiil, U. K., and Leggett, J. J. HyperForm: using extensibility to develop dynamic, open and distributed hypertext systems. In *ECHT '92. Proceedings of the ACM conference on Hypertext, November 30-December 4, 1992, Milan, Italy* (1992), pp. 251–261.
25. Wiil, U. K., and Østerbye, K., Eds. *Proceedings of the ECHT '94 Workshop on Open Hypermedia Systems* (1994). Technical Report R-94-2038, Dept. of Computer Science, Aalborg University.

This article was processed using the LaTeX macro package with LLNCS style