Extreme Markup Languages 2002

Montréal, Québec
August 6-9, 2002

Interoperability Between Collaborative Knowledge Applications

Eugene Eric Kim
Independent Consultant

Ken Holman
Crane Softwrights Ltd.

Abstract

The purpose of Doug Engelbart's Open Hyperdocument System (OHS) is to enable applications to implement sophisticated hyperlinking functionality in an interoperable manner. In order to achieve this, there must be a standard architecture for representing hyperdocuments. Experience suggests that such an architecture be based on a graph-based data model, consisting of nodes with properties and typed links. In this paper, we briefly introduce the OHS, and explain the advantages of a graph-based data model, exploring the notion of a hyperdocument. Using a threaded forum discussion as an example, we show how to represent a hyperdocument using a graph-based data model, and how to integrate different hyperdocuments in useful and compelling ways.

Interoperability Between Collaborative Knowledge Applications

Table of Contents

Interoperability Between Collaborative Knowledge Applications

Eugene Eric Kim and Ken Holman

§ 1 Introduction

Doug Engelbart's NLS — the original hypertext system, which was developed in the 1960s -- had a marked advantage over today's knowledge management systems: It was the only one. All of the software applications used in Engelbart's lab — document editors, software development tools, e-mail, and so forth — were subsystems of NLS, and hence, all of these applications could interoperate with each other. Users could apply a single dynamic view to any type of document. Programmers could link relevant e-mails, such as design discussions or bug reports, to sections of source code.

Today, we have a far greater breadth of knowledge applications at our disposal, ranging from the universal (such as word processors and e-mail) to the domain-specific. Many of these applications share some level of functionality, including some of the hypermedia functionality found in the original NLS. However, most of these applications can only interoperate with each other in very limited ways. As a result, today's systems are incapable of doing many of the things that were possible in Engelbart's lab 40 years ago.

Recognizing the importance of interoperability between knowledge applications and the lack of progress in that regard, about 15 years ago, Engelbart proposed the Open Hyperdocument System (OHS), an open infrastructure for collaborative knowledge management applications. [Engelbart 1990] [Engelbart 1992] The OHS's functional requirements largely mirror the feature set of NLS and its commercial successor, Augment. [Engelbart 2000] However, the main goal of the OHS is not to build yet another hypermedia system. Instead, it aims to enable any knowledge application to implement these features in an interoperable manner. [Engelbart 1990]

In order to enable sophisticated hyperlinks and relationships between data contained in a variety of document formats, there needs to be a standard architecture for hyperdocuments. Experience suggests that such an architecture be based on a graph-based data modeling language, consisting of nodes and typed, directional links.

The SGML/HyTime and World Wide Web and Topic Maps communities have already done much work to show how a graph-oriented language (respectively Groves, RDF, and Topic Maps) can be used to model familiar, document-oriented content, such as articles stored as SGML or Microsoft Word documents. In this paper, I further explore how hyperdocuments may be modeled using a graph-oriented language, and the implications of doing so. I then explain how such a data modeling language may serve as the basis for a standard hyperdocument architecture. Finally, I describe the current status of the OHS.

§ 2 OHS Requirements

The OHS's requirements are well-documented [Engelbart 1990] [Engelbart 2000] and well-understood in the hypermedia community. Nevertheless, for the purposes of this paper, it is useful to review some of them.

Addressability is the fundamental requirement upon which all other hyperlinking capabilities depend. [Prescod 1999] Perhaps the most important OHS requirement is *high-resolution addressability*, also known as "granular addressability," "fine-grained addressability," and "object-level addressability." The World Wide Web offers the capability to link to documents and some limited capability to link to specific parts of an HTML document. High-resolution addressability provides the capability to link to objects that are more granular than a document, such as paragraphs in a textual document, pixels in a bitmapped image, and frames in a video.

Two requirements related to hyperlinks are *back-link management* and *typed links*. Most hypermedia systems, including the World Wide Web, show links from an object. Back-link management allows the additional ability to show links to an object. Typed links essentially enable you to label links. For example, a document that is annotating another document could be linked to the annotated document using the type "annotation."

Another important requirement is the ability to *specify views* for displaying hyperdocuments. For instance, you could specify a view that would only display the headers of an article, or the first line of every paragraph. You could also specify the view to use when traversing a link.

§ 3 Graph-Based Data Models and Hyperdocuments

The goal of the OHS is to enable knowledge applications to implement these requirements interoperably, independent of factors such as document format. There are two approaches to enabling such interoperability. The first is to standardize a single data model, and develop technology that uses it. This is the approach that W3C has taken with XML and its **http://www.w3.org/TR/xml-infoset/** Infoset data model. W3C standards, such as XSL and XLink, apply to any documents that conform to the Infoset data model.

The second is to standardize a metalanguage for defining data models, which is the approach of SGML/HyTime Groves. By expressing data models for different documents using the same set of constructs, technology could be developed to work on those constructs rather than on one particular data model. By standardizing the data modeling language rather than the data model, the OHS may work with a variety of document formats interoperably.

Experience suggests that a graph-based data modeling language consisting of nodes and links is the most appropriate way to express knowledge containers for the purpose of hyperdocument applications. The SGML/HyTime community has already done much work to show how a graph-oriented language can be used to model familiar, document-oriented content, such as articles stored as SGML or Microsoft Word documents. [Kimber et al 2001] In this paper, I explore how hyperdocuments may be modeled using a graph-oriented language, and the implications of doing so.

The notion of a document is largely influenced by our paper-based world. Documents are thought of as discrete packages of information, and are generally stored and represented that way on filesystems. Unfortunately, this file-based paradigm has constrained our thinking as well as our capabilities. Most operating systems share a very similar API for creating and manipulating files, which in turn represent documents. The aim of the OHS is a common API for creating and manipulating hyperdocuments.

Hyperdocuments are often expressed using two separate data models: one that represents documents, and one that represents the relationships between documents. XML Infoset and SGML/HyTime Groves are examples of the former, whereas RDF, Topic Maps, and XLink are examples of the latter. The reason for the separation is that structural links within documents are thought to be different from links that represent relationships between documents, a consequence of our paper-based, document-centric paradigm. This can be a valid distinction, but there is no reason that the different data models could not be expressed using a common data modeling language. Infoset, Groves, RDF, Topic Maps, and XLink are all graph-based representations that share a paradigm of nodes with properties and typed, directional links. A typed link could just as easily represent document structure as it could relationships between documents. For example, RDF could be used to model documents as well as the relationships between documents.

§ 4 Case Study: Threaded Forum Applications

A single metamodeling language for representing hyperdocuments could be used as the basis of an API for creating and manipulating hyperdocuments. This API, in turn, could enable all types of knowledge applications to interact with each other in useful and compelling ways.

The ubiquitous threaded forum application is perhaps the most commonly used tool for collaborative, online discussions. As such, it is probably the largest and most poorly organized source of shared, stored knowledge on the Internet today. Threaded forums applications were designed for linear, asynchronous, minimally-structured discussions, not for generating well-organized knowledge repositories. There may be a wealth of useful knowledge stored in these archives, but the only way you could find out would be to read each message.

Tool developers have attempted to overcome the limitations of threaded forum applications by adding new features. These features themselves are usually limited in that they only work in certain applications and that they sometimes conflict with the original features that made the applications compelling in the first place.

Expressing the threaded forum application's data as hyperdocuments using the OHS's data modeling language helps overcome these limitations. Tools designed to organize or manipulate data expressed using this graph-based data modeling language could be applied towards threaded forum data. For example, you could use a scheme for categorizing Web sites, such as Topic Maps, to organize threaded forum posts as well. You could apply a stylesheet that was written to reformat selected paragraphs from an XML document to generate a version of selected excerpts from a threaded discussion.

Thinking about threaded forum software in OHS terms is a good way to show how the OHS benefits collaborative knowledge applications. A threaded forum consists of a series of discrete messages, some of which may be responses to other messages. Examples include mailing lists, USENET newsgroups, and Web-based discussion forums such as **http://slashdot.org/** SlashDot.

A data model for threaded forum applications might look like Figure 1.

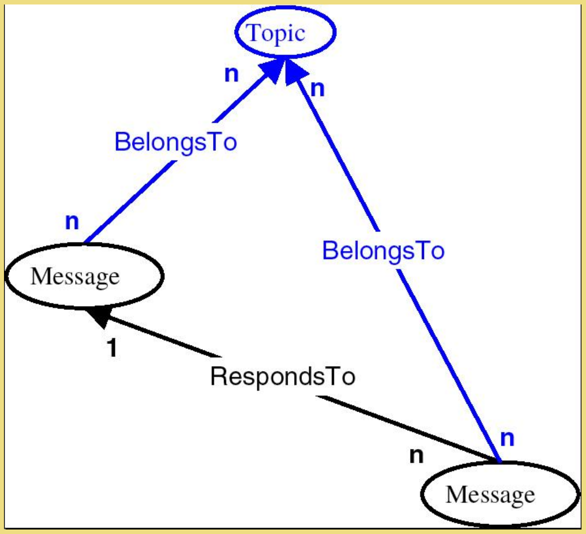


Figure 1

Each message is a discrete entity that falls under some topic. The topic could be a message board topic or the name of the mailing list or newsgroup. A message may be posted to multiple topics. A message may respond to another message. Messages may have multiple responses; however, a message may only respond to one other message.

From an OHS perspective, messages and topics are all part of the same hyperdocument, which is represented by the graph in Figure 1. The ID of the message to which it is responding is considered a link of type "RespondsTo."

As a more concrete example, consider the following exchange on the ba-ohs-talk@bootstrap.org mailing list. Alice sends a message to the list:

```
Message-Id: 1
From: alice@foo.com
To: ba-ohs-talk@bootstrap.org
Subject: OHS?

What is the OHS?
```

Bob sends a different message to the list:

```
Message-Id: 2
From: bob@bar.com
To: ba-ohs-talk@bootstrap.org
Subject: graphs

Graph-based data models are the way to go, because they can usefully
represent all kinds of data.
```

Meanwhile, Carl responds to Alice's message:

```
Message-Id: 3
From: carl@junior.com
To: ba-ohs-talk@bootstrap.org
In-Reply-To: 1
Subject: Re: OHS?

> What is the OHS?

An infrastructure that enables collaborative knowledge applications
to interoperate.
```

This exchange can be modeled using the data model from Figure 1, as depicted in Figure 2.

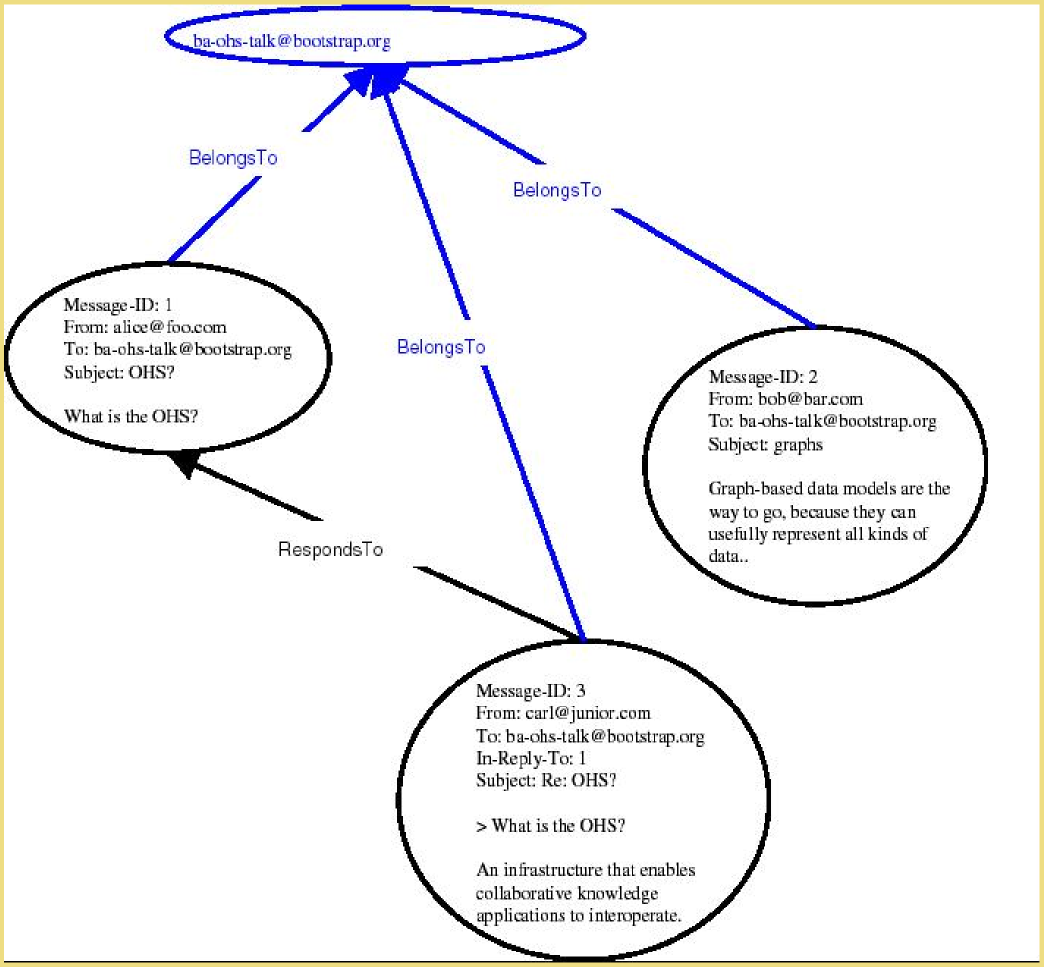

ba-ohs-talk@bootstrap.org
BelongsTo
BelongsTo
BelongsTo
RespondsTo
Message-ID: 1
From: alice@foo.com
To: ba-ohs-talk@bootstrap.org
Subject: OHS?
What is the OHS?
Message-ID: 2
From: bob@bar.com
To: ba-ohs-talk@bootstrap.org
Subject: graphs
Graph-based data models are the way to go, because they can usefully represent all kinds of data..
Message-ID: 3
From: carl@junior.com
To: ba-ohs-talk@bootstrap.org
In-Reply-To: 1
Subject: Re: OHS?
> What is the OHS?
An infrastructure that enables collaborative knowledge applications to interoperate.

Figure 2

Threaded forum applications generally offer multiple ways to view a discussion — in other words, multiple view specifications. You can sort by author, date, or thread. In the case of SlashDot, you may view all of the contents of the thread on one page, either as a flat list or a nested list.

A thread view of the above exchange might look like:

```
* OHS?, Alice
    * Re: OHS?, Carl
* graphs, Bob
```

In order to generate this display, the application must know which messages point to a particular message. In other words, it needs access to back-link information. Applications will either store link information in the headers of messages, as in the case of mailing lists and USENET newsgroups, or in a back-end database, as in the case of most Web-based forums. In the former case, applications must generate a link database by parsing the headers of all the messages. In the latter case, a link database already exists.

There are two important reasons why a graph-based data model (depicted in Figure 1) is a useful way of thinking about threaded forum applications.

First, because the abstraction applies equally well on all threaded forum applications, you can develop data repositories with a standard interface for handling these types of discussions. If you then refactor existing applications to use one of these data repositories on the backend, then all of these applications will be able to interoperate with each other's data. This would enable the following scenarios:

- View and participate in a discussion hosted on a Web-based forum via e-mail. Your contributions would be visible both on the Web-based forum and from your e-mail reader.
- Respond to a thread on SlashDot on your own Web-based forum. Have your response be seamlessly integrated into the SlashDot threaded view, even though it is hosted on your system.
- Replace your forum application with a different tool, while retaining your legacy data *in its current form*. You would not have to import the data into the new tool, because you could access the old data in the old tool using the standard interfaces.
- Visualization tools could be refactored to use this interface as well, so that they could be used to visualize threads in any tool. Warren Sack has developed a tool called **http://www.sims.berkeley.edu/~sack/CM/** Conversation Map, which constructs useful visualizations of discussion threads from USENET newsgroup and mailing list discussions. If Conversation Map were modified to use a standard interface for retrieving message and back-links, it could be used to construct visualizations of any tool that also used that interface.

Second, using a graph-based data model would allow you to integrate threaded forum discussions with other types of data that are also represented using graph-based data models.

4.1 *Dialog Mapping*

Dialog Mapping tools use a grammar called **http://www.gdss.com/wp/IBIS.htm** IBIS, which allows you to structure discussions in useful ways. The resulting map generally depicts an organized, easy-to-follow snapshot of discussion.

IBIS consists of questions (i.e. issues), ideas (i.e. responses to questions), and arguments (i.e. pros and cons of ideas). Figure 3 is a data model for an IBIS discussion.

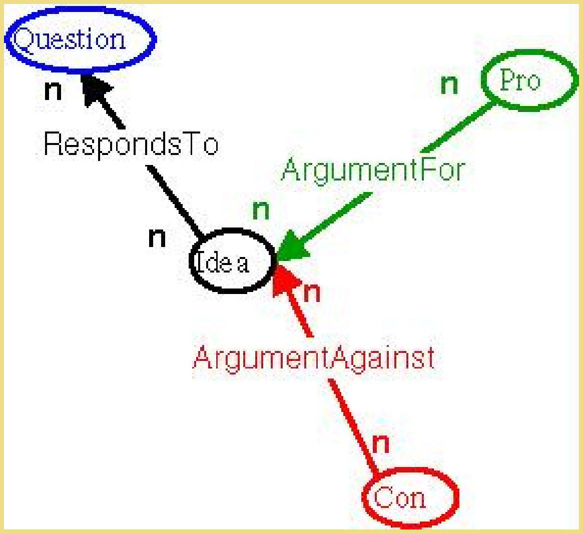


Figure 3

Software that support Dialog Mapping and IBIS include **http://www.softbicycle.com/questmp.html** QuestMap, **http://www.compendiuminstitute.org/tools/mifflin.htm** Mifflin, and **http://www.r-objects.com/products/** Pepper. Figure 4 shows a sample Dialog Map discussion using Mifflin.

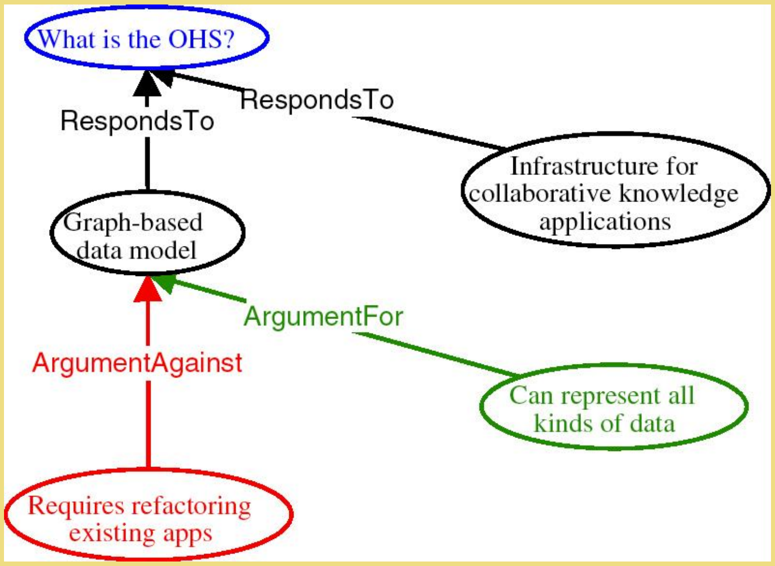

What is the OHS?
RespondsTo
RespondsTo
Graph-based data model
Infrastructure for collaborative knowledge applications
ArgumentFor
ArgumentAgainst
Can represent all kinds of data
Requires refactoring existing apps

Figure 4

Figure 5 is this same discussion depicted using the data model in Figure 3.

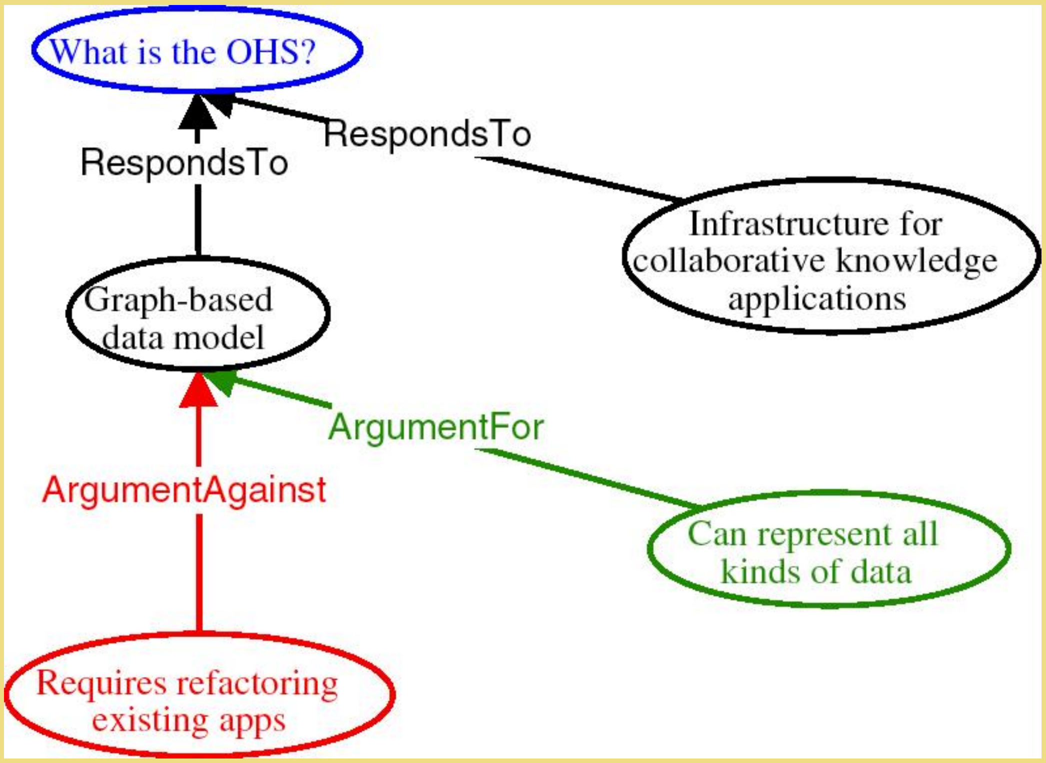

What is the OHS?
RespondsTo
RespondsTo
Infrastructure for collaborative knowledge applications
Graph-based data model
ArgumentFor
ArgumentAgainst
Can represent all kinds of data
Requires refactoring existing apps

Figure 5

Note how easily the Dialog Map maps to the graph-based data model. This is by design, as these tools were designed with hypertext in mind.

4.2 *Integrating Dialog Maps with Threaded Discussion*

In the threaded forum example above, there is a discussion about the OHS. While the forum is a comfortable way for participants to interact, the threaded view doesn't reveal much about the content of the discussion.

The subsequent Dialog Map example presents a much clearer picture of the content of the discussion and could easily serve as a method for organizing the discussion in the threaded forum. Every instance of some concept discussed in the threaded forum should be related to the equivalent IBIS concept.

In order to do this, you could create a new link type, "OccurrenceOf", which would map any IBIS node to any message node. Figure 6 shows the integrated data model.

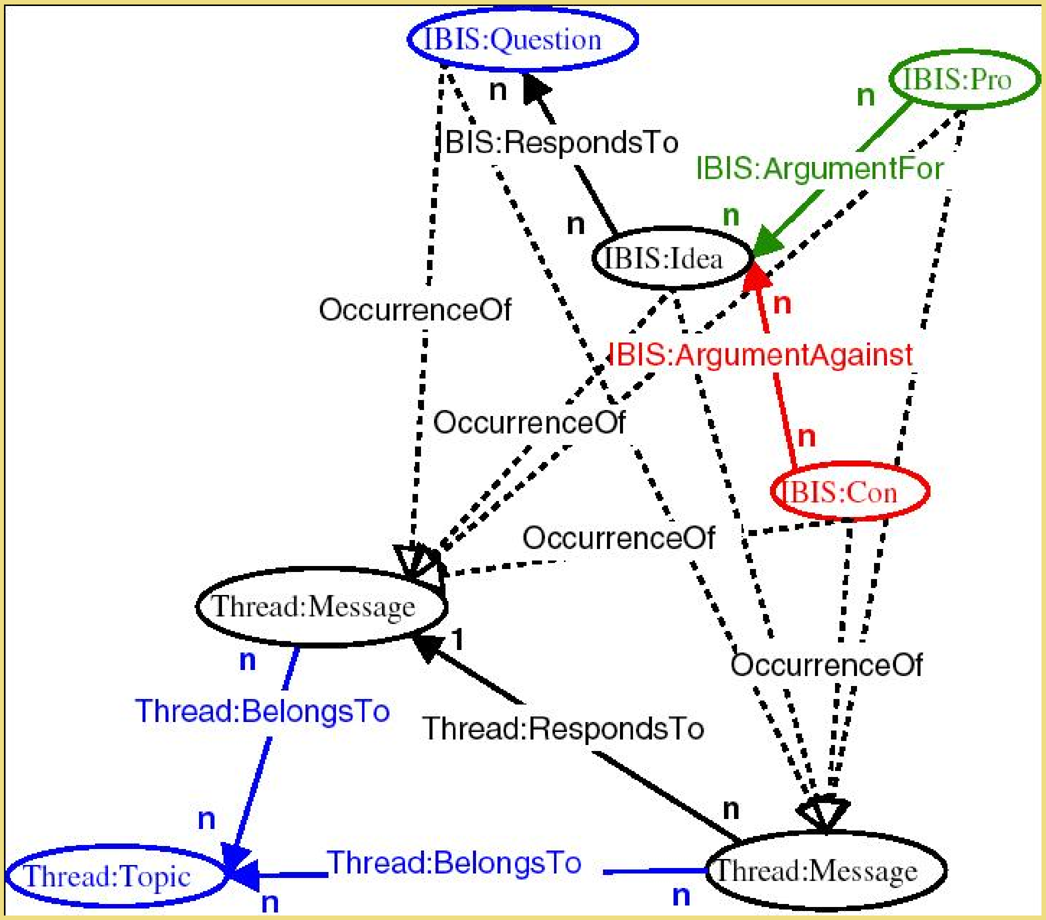
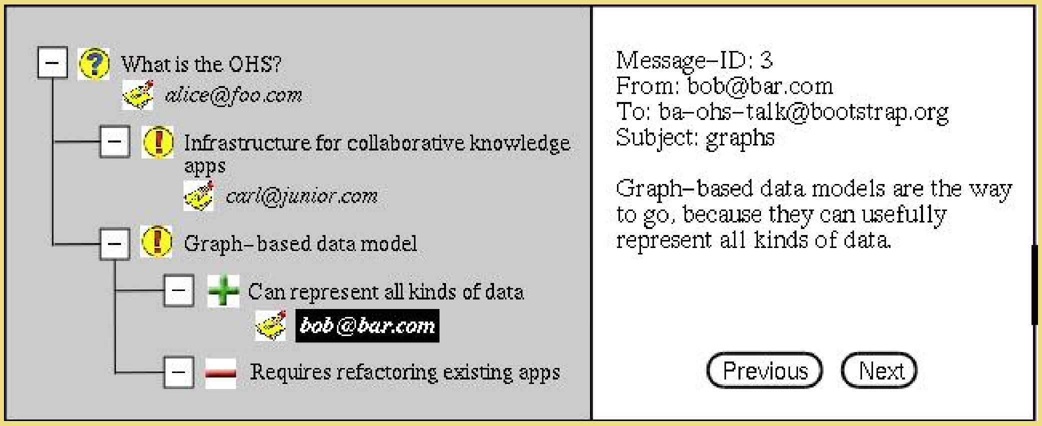


Figure 6

As you can see, this combined data model uses namespaces — "IBIS" and "Thread" — in order to differentiate between the two different data models.

The two discussions about the OHS map easily to each other using this data model. Alice's e-mail is an occurrence of the IBIS question, "What is the OHS?", so the two would be linked using an "OccurrenceOf" link. Bob's e-mail is an occurrence of the IBIS argument in favor of graph-based data models. Carl's e-mail is an occurrence of the IBIS idea, "Infrastructure for collaborative knowledge apps."

Relating these two sets of discussions in this way creates a new, useful context for the content, which enables new ways of visualizing the content. Figure 8 shows a forum message in the context of the Dialog Map of that discussion.



What is the OHS?
alice@foo.com
Infrastructure for collaborative knowledge apps
carl@junior.com
Graph-based data model
Can represent all kinds of data
bob@bar.com
Requires refactoring existing apps

Message-ID: 3
From: bob@bar.com
To: ba-ohs-talk@bootstrap.org
Subject: graphs

Graph-based data models are the way to go, because they can usefully represent all kinds of data.

Figure 7

Allowing the integration of different types of data enables a whole new class of useful views, applications, and features, while allowing you to retain existing applications and data.

§ 5 Conclusions

The examples above indicate that a graph-based data model is a useful way to represent hyperdocuments. The experiences of the document modeling and knowledge representation communities confirm it.

Not surprisingly, there are a number of technologies that allow you to do graph-based modeling, and all of them are viable candidates for the OHS. RDF and Topic Maps are two such tools. RDF was designed as a knowledge representation language, but because its triples data model is so general, it could be used to model different knowledge containers, as described above. Topic Map, which also uses a triples model, is another viable candidate for the same reason. Two other candidates are **http://www.concept67.fsnet.co.uk/gsix/** GSIX and **http://www.gupro.de/GXL/** GXL. Both of these were designed as graph-interchange languages, and both have well-defined data models.

The best candidate thus far is the aforementioned SGML/HyTime Groves, which seems to meet the basic requirements of a data modeling language for the OHS. Doug Engelbart is currently in the process of putting together a community and a process for evaluating the various candidates. However, choosing a data modeling language is only a first step in the design and implementation of the OHS, and it cannot be done without considering implementation issues. Groves, for example, uses nodes to define the granularity at which a document may be addressed. If you want character-level addressability of a document, you must define a node for each character. However, the OHS must have a higher-level view of a node. Lee Iverson, who has proposed his Groves-inspired NODAL system [Iverson 2001] as the data modeling language for the OHS, defines nodes as the fundamental units of metadata within the system, which includes an immutable ID, versioning information, and access control privileges.

Issues such as these must be carefully examined before a data modeling language can be finalized, and work on the OHS may progress. In order to evaluate these issues properly, the community needs open source implementations of these candidate technologies. This is the biggest current obstacle towards building an OHS.

§ 6 Acknowledgements

Many thanks go to Murray Altheim, Eric Armstrong, Dennis Hamilton, G. Ken Holman, Lee Iverson, Peter Jones, Sandy Klausner, Jack Park, Alex Shapiro, Henry van Eyken, and Joe Williams for their insightful feedback. Special thanks goes to Doug Engelbart, the inspiration for this work, and a mentor to us all.

§ 7 Bibliography

[**Engelbart 1990**] Engelbart, Douglas C. "Knowledge-Domain Interoperability and an Open Hyperdocument System." *Proceedings of the Conference on Computer-Supported Cooperative Work*, Los Angeles, CA. October 7-10, 1990, pp143-156. **http://www.bootstrap.org/augdocs/augment-132082.htm** http://www.bootstrap.org/augdocs/augment-132082.htm

[**Engelbart 1992**] Engelbart, Douglas C. "Toward High-Performance Organizations: A Strategic Role for Groupware." *Proceedings of the GroupWare '92 Conference*, San Jose, CA, August 3-5, 1992, Morgan Kaufmann Publishers. **http://www.bootstrap.org/augdocs/augment-132811.htm** http://www.bootstrap.org/augdocs/augment-132811.htm

[**Engelbart 2000**] Engelbart, Douglas C. "Draft OHS-Project Plan." October 23, 2000. **http://www.bootstrap.org/a2h/BI/2120.html**

[**Iverson 2001**] Iverson, Lee. "NODAL: A Filesystem for Ubiquitous Collaboration." September 20, 2001. **http://nodal.sourceforge.net/NODAL-WhitePaper.html**

[**Kimber et al 2001**] Kimber, W. Eliot, Mark Anderson, and Brandon Jockman. "XSL and Hyperdocuments: Applying XSL to Arbitrary Groves and Hyperdocuments." *Extreme Markup Languages 2001*, Montreal, Canada, August 14-17, 2001, pp85-123. **http://www.isogen.com/papers/XSLandGroves.html**

[**Prescod 1999**] Prescod, Paul. "Addressing the Enterprise: Why the Web needs Groves." July, 1999. **http://www.prescod.net/groves/shorttut/**

The Authors

Eugene Eric Kim
Independent Consultant
63 Bovet Road, PMB 207
San Mateo
California
United States of America
eekim@eekim.com
http://www.eekim.com/

Eugene Eric Kim is a freelance writer, programmer, and consultant. He has written for a number of publications, from Outdoor Retailer to Scientific American. Previously, he served as the Senior Technical Editor at Dr. Dobb's Journal, the leading magazine for software developers. Eugene is the author of CGI Developer's Guide (Sams.net 1996), and is currently writing a book on the history of free software. He received his A.B. in History and Science from Harvard University.

Ken Holman
Crane Softwrights Ltd.
Kars, Ontario
Canada
gkholman@CraneSoftwrights.com
http://www.CraneSoftwrights.com

Mr. G. Ken Holman is the Chief Technology Officer for Crane Softwrights Ltd., a Canadian corporation offering OmniMark programming, DSSSL and XSL/XSLT language training, and general SGML and XML related computer systems analysis services to international customers. Mr. Holman is the current Canadian chair of the ISO subcommittee responsible for the SGML family of standards, the current chair of the OASIS XSLT/XPath Conformance Technical Subcommittee, an invited expert to the W3C, the former chair of the OASIS XML Conformance Technical Subcommittee, the author of electronically-published and print-published books on XML-related technologies, and has often been a speaker at related conferences. Prior to establishing Crane, Mr. Holman spent over 13 years in a software development and consulting services company working in the NAPLPS and the SGML industries.

Extreme Markup Languages 2002
Montréal, Québec, August 6-9, 2002
This paper was formatted from XML source via XSL
Mulberry Technologies, Inc., August 2002