Navigating Spaces : The Semantics of Cross Domain Interoperability

David Millard and Hugh Davis

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

Abstract. The Open Hypermedia Protocol (OHP) developed by the Open Hypermedia Systems Working Group (OHSWG) allows components of a system to discuss Navigational Hypermedia, but it does not yet address the needs of other hypertext domains. The focus of Structural Computing has been to recognise the need for structure at all levels of computing, if OHP could express the structure of multiple domains then it could be used to facilitate this goal. The Fundamental Open Hypermedia Model (FOHM) has been developed to incorporate other domains into the OHP data model, forming a semantic language that can potentially be used to discuss any structure. In this paper we look at the 'cross-domain fertilization' that takes place when several domains are brought together in this way and describe what it means to encounter structures from one domain in another.

1 Background

The Open Hypermedia Systems Working Group (OHSWG) has been working on the problem of interoperability between hypermedia systems and has produced the Open Hypermedia Protocol (OHP), an evolving standard by which such systems can communicate. Originally this work focused on Navigational Hypertext, but recently the group has been attempting to extend OHP to apply to other Hypertext Domains (such as Spatial and Taxonomic Hypertext). Focus has also shifted to the paradigm of structural computing, where the issues of Navigational Hypermedia have been recognised as part of a wider problem domain, that of managing structure throughout a computer system.

The authors have been working on the **F**undamental **O**pen **H**ypertext **M**odel (FOHM), which extends the OHP data model to apply to these other domains. FOHM considers each domain as a particular way of using the same basic associational structures. Combining all three views results in a model that could be used as an exchange format between components of differing domains and could theoretically form a basis for multi-domain knowledge transfer in future systems.

Because FOHM encompasses the functionality of all three domains, when using the model in any particular domain one must handle the structures normally unique to the others. Initially this was seen as a problem as it introduced unnecessary complexity into each domain, but surprisingly it has resulted in each domain growing in a useful manner. We have described this as *cross-domain fertilisation*. In essence FOHM has become more than the sum of its parts.

In this paper we explore this cross-domain fertilization and look at what structures are supported in FOHM as a result.

2 An Overview of Hypertext Domains

In its work on interoperability the OHSWG has been considering the requirements of several domains. The three most frequently mentioned were Navigational, Spatial and Taxonomic Hypertext. Before we can examine FOHM it is necessary to define these domains.

2.1 Navigational Hypertext

Navigational Hypertext is the most traditional domain of hypertext, exemplified in Open Hypermedia Systems such as Chimera [1], DHM [5], HyperForm [15], Microcosm [3] and the HB/SP series [13]. It is also used by the World Wide Web (WWW) [2] which has resulted in a generation of people becoming comfortable with the point and click interface commonly associated with it.

FOHM is strongly based on the OHP data model of navigational hypertext [4], demonstrated by Figure 1.

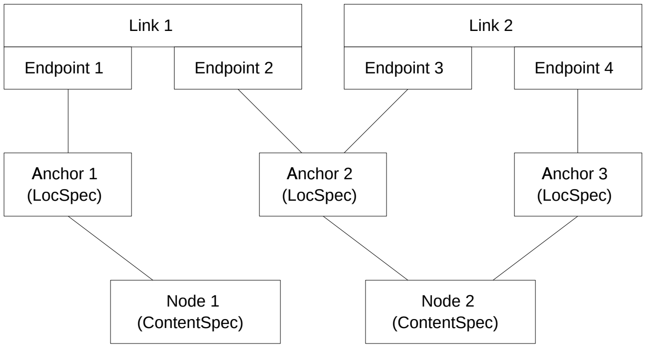

Fig. 1. The OHP Node Link model

A link is a defined relationship between zero or more endpoints, where each endpoint references an anchor. These are objects that attach the link to other objects, typically nodes (wrappers for documents). An anchor can attach either to the whole object or to a location inside that object. Multiple anchors can attach to the same thing and multiple endpoints can reference the same anchor.

The endpoint contains information that is separate from the anchor itself but is useful for a link. Typically this is a direction attribute, such that in any link an endpoint may be a source, destination or bidirectional endpoint. This is then used in the Follow Link operation that a user invokes to navigate the structure.

2.2 Spatial Hypertext

Spatial Hypertext systems allow users to organise their information visually in a process known as ‘Information Triage’ [7]. Relationships between nodes are expressed by their visual characteristics such as proximity, colour or shape. This results in some interesting properties. If for instance a node is slightly misaligned with other nodes then this might express an uncertainty about whether the node is actually part of the relationship. In other words it expresses classification within relationships, where some nodes are ‘more’ related then others.

Spatial hypertext systems are therefore ideal for an evolving organisation of data. Examples of such systems include VIKI [6] and CAOS [12].

2.3 Taxonomic Hypertext

Taxonomic Hypertext is the categorisation of information, called artifacts, into sets. Applications can allow users to navigate the information space by moving between overlapping sets and can also reason about the relationships that nodes have with one another, represented by the sets. In addition it expects there to be differing views on how the information is to be categorised and defines perspectives, objects that allow the categorisation hierarchy to split according to the views of differing users.

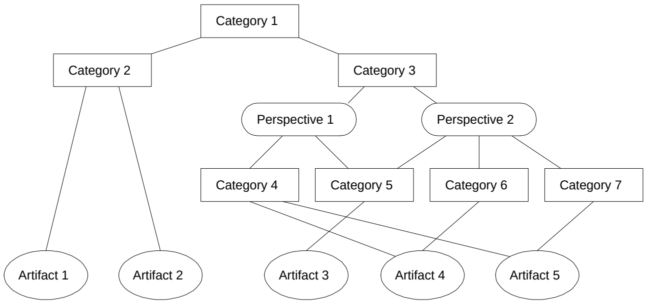

Fig. 2. Example Taxonomy

Figure 2 gives an example of a Taxonomy. Here two people have categorized five artifacts. They both agree that all five lie within category one and that artifacts one and two also lie within sub-category two. However they disagree on how the remaining three artifacts should be split up. The two perspective objects represent the branching point of the categorisation tree.

3 A Common Model

3.1 FOHM and Structural Computing

FOHM makes no assumptions about the protocol it is running over or the systems that are using it. It is a semantic language that requires an implementation in a syntactic language before it can be used. We currently have one implementation of FOHM running across the SoFAR agent framework developed at Southampton [9]. In this implementation FOHM is described in an ontology which is then discussed over the agent infrastructure.

Structural Computing "asserts the primacy of structure over data" [10], as such it is concerned with looking at how structure can be discussed and managed at all levels of computing. FOHM is a semantic language for the discussion of structure. Although other structural languages exists they tend to contain very little semantic information (e.g. XML) and suffer performance penalties as a result of their generality [11].

3.2 A Description of FOHM

In FOHM we describe four objects that are analogous to objects in the OHP-Nav data model [8]. An *association* contains a feature space; a list of features that all the objects in the association must map to. It also contains a set of *bindings*, these attach *datarefs* to the association via a feature vector that describes how the dataref maps to the feature space. Finally FOHM has a notion of a *data* object, this is a wrapper for some piece of data that lies outside the scope of the model, normally a document although it could represent any file or stream.

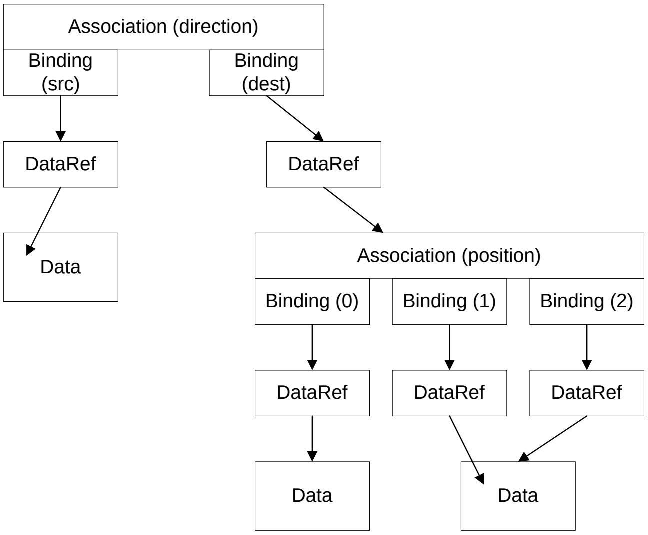

Fig. 3. FOHM Structures: A Navigational Link to a Spatial List

Figure 3 shows a possible FOHM structure. Bindings map DataRefs to the Navigational Link on the left by defining their direction and to the Spatial List on the right by defining their position. The DataRefs either reference a whole item of Data or point into that Data (e.g. to reference a particular region).

3.3 Levels of Structure in FOHM

Between them the domains modelled support three levels of structure:

1. *Explicit External Structure*. By creating typed associations between data, FOHM allows explicit relationships to be expressed. This is analogous to taxonomic categorisation, where we say that an object (or data) belongs in one category or another.
2. *Implicit External Structure*. This is classification within a relationship, defined by the feature space of an association and the corresponding feature vectors. E.g. in Spatial Hypertext an association may contain a set of data objects, each one of which is given a Red/Green/Blue (RGB) value. Objects which are similar in some way will be more alike in colour.
3. *Internal Structure* is the structure of the information actually inside individual data objects. For example a film can be viewed as a collection of many scenes, a user viewing the film follows a path or trail through those scenes. Typically this is not the way a film is stored due to file size and performance restrictions. Thus at some level we no longer handle structure externally and instead handle it internally via a proprietary data format. Datarefs allows FOHM's external structures to reference the internal structure of data (e.g. link to the seventh scene of a film).

One of the most important aspects of FOHM is that if a client does not totally understand the semantics of the structure it is given it may still understand a portion of it. For example imagine an association that represents a company. This association may have a feature space with a single feature 'role'. All data that binds to this association must have a vector that maps to that feature, in effect stating what the role of that data object is in the company.

Should this structure be served to a client that understands the feature space 'role' then that client will appreciate all of the meaning of that structure. However if it was served to a client that didn't understand the feature then that client would at least still understand that a relationship exists, could display the relationship to the user and otherwise manipulate the association as normal.

4 The Contribution of the Domains

Each domain brings something unique to the FOHM model. In this section we will look at what is brought to FOHM from each domain and discuss how each of the domains are extended to handle the extra structure.

4.1 Navigational Hypertext: Introducing the Anchor

In many ways Navigational Hypertext is the simplest of the three domains. Its notion of directed links can be easily modeled in FOHM by a single feature 'direction' to which nodes are bound with either a 'source', 'destination' or 'bi-directional' value. The anchor object allows external linking structure to point into otherwise opaque data, it is this mechanism that allows the referencing of *internal structure* as mentioned above. However neither Spatial or Taxonomic Hypertext has such a mechanism.

In FOHM the anchor is replaced by a dataref object. This object is also accessible by Spatial and Taxonomic clients enabling those domains to reference internal structure as well. This is useful to both domains and allows spaces and categories to refer to parts of a data object as well the object in its entirety.

4.2 Spatial Hypertext: Classification within Relationships

At a superficial level it is possible to view Spatial Hypertext as a presentation layer on top of Navigational Links. For example a link may have some attributes that determine that when viewed spatially it appears as a red square. However this is to miss the important notion of *implicit external structure* as described above. In a Spatial Hypertext System the visual attributes of the various objects actually form extra structural information about those objects, one of the applications of which is to allow fuzzy membership of an association.

Spatial Hypertext Systems rely on users to understand the visual clues supplied. E.g. they can express that one object is redder than another but no semantic reason is given, the user must interpret that information themselves. In FOHM we use the feature space to contain all the spatial features that datarefs may bind to, e.g. colour, shape, size, etc. A binding maps values to those features, therefore describing a datarefs position in the space. An interesting consequence of this approach is that as FOHM has no restriction on the feature spaces used it is possible to replace spatial mappings with semantic ones. In effect allowing the system, as well as the user, to appreciate fuzzy membership.

As an example consider a set of nodes that represents dangerous animals. In a Spatial Hypertext system we may colour the animals such that red indicates danger. When rendered the user can see what the nodes represent and they can also see that some are redder than others, but the understanding of what red represents is lost. However in FOHM we could replace the colour with a different feature that explicitly defined that semantic, e.g. ‘danger’, which has defined values ranging from ‘harmful’ to ‘deadly’. A system that understands the meaning of the ‘danger’ feature has a true understanding of how the animals relate to one another. The disadvantage of this is that it is probable that more systems understand colour than a specific feature like ‘danger’.

The feature space is a very powerful way of binding objects to an association, however we must extend Navigational and Taxonomic Hypertext to deal with these powerful bindings.

With Taxonomic Hypertext this is a fairly trivial operation, the *implicit external structure* simply becomes an extension of the categorisation process, where the fuzzy membership of a category becomes possible. In addition if the people creating a taxonomy disagree on this implicit structure then they can divide the taxonomy at that point using the perspective feature as normal.

Navigational Hypertext has many more issues to deal with because it relies on the feature space to enable the navigation of associations. There are two ways to cope with the 'pollution' of the feature space.

1. Ignore it. A Navigational client can assume that if there is no direction feature then all members of an association can be treated as bi-directional for purposes of navigation.
2. Extend the navigation model. We can make the same assumption as above but also extend our model of navigation so that it understands some of the common features that would be used by the other domains and allows them to alter the effect of navigation.

In actual fact this is a decision that is beyond the scope of the model, however it is important to think about the consequences of a client taking the second approach. If we had a standard definition of what it means for a client to navigate a list (as opposed to a link) then hypertext designers could build their hypertexts to take advantage of that functionality.

There are two places were the structure of an association could become important. The first is during *traversal*, applying a Follow Link operation to an association, the second is *arrival*, where following a link results in another link, rather than a document. This behaviour remains beyond the scope of the model, but still requires definition if interoperability is to be achieved at a useful level. Some example definitions are given in [8].

4.3 Taxonomic Hypertext: Perspectives and Context

Although the OHP-Nav data model has no notion of context, context has long been an important issue in navigational systems, allowing a user to see different versions of documents or hyperwebs according to a particular viewpoint. In Taxonomic Hypertext context is realised via the use of a perspective object. These are designed to be placed in a categorisation hierarchy at the point where it splits according to the views of the authors. The context of the viewer will determine which perspective (branch of the taxonomy) that they see.

When implementing a perspective in FOHM one could use an association of type 'perspective'. However this does not fit in very well with the rest of the model. This is because the semantics of what a perspective does to traversal and arrival functionality has to be understood by the association containing those perspectives and not by the perspectives themselves. In addition what would it mean for that parent association to have structure? What is the meaning of a list of perspectives as opposed to a set?

Fortunately there is a way of shifting this knowledge back into the perspective object itself. Rather than a category containing perspectives representing a branch, in FOHM we say that a perspective object *is* the branch. If there were two versions of a data object you would create a perspective association that contained both versions. Now the arrival semantic of a perspective is to choose one of the data objects to reveal and the traversal semantic is essentially the question 'what other views are there on this object?'

Figure 4 shows these alternative implementations of perspective. Note that in the FOHM model there are two versions of Category 1 - the result of viewing that Category in different contexts.

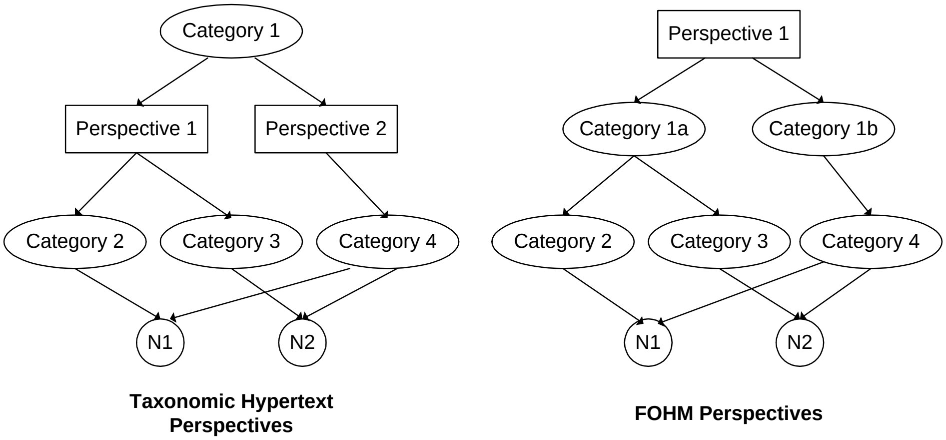

Fig. 4. Alternative implementations of perspective

The FOHM model of perspective contains some properties that are very desirable when dealing with contextual systems:

1. The ability to link to a specific node, whatever the contextual considerations (linking across context). This is achieved by linking to the node itself.
2. The ability to link to a node that is determined by context. This is achieved by linking to the perspective that contains the choice of nodes.
3. Everything about the structure can change in context. Including the type of that object! (e.g. linking to a perspective could result in a data object in one context but result in an association object in another)

However FOHM only gives a framework in which context can operate, it does not define context itself. A context is an object that contains contextual information. FOHM assumes the existence of two types of context.

1. *User Context*. This is an object that defines the state of the user. This could include information about the user (such as age and job position) as well as information relating to their interaction with the system (trails etc.). In addition it may include temporary user specified information that will help the user filter the information space (e.g. today I am interested in cars).
2. *Data Context*. This is an object that describes a data object. This could include information about the object (such as the date of its creation), versioning information as well as a description of to whom the object would be interesting.

In addition FOHM assumes the existence of a 'magic function' that compares a user context to a data context and decides if they match. A simple implementation could define a the contexts as lists of keyword value pairs and then the function would choose an object based on the number of matching values.

5 Conclusions

FOHM is a semantic language in which associational structures can be discussed. In this paper we have shown how those structures draw from the three original domains that FOHM was based on. Enabling structure to be represented at many levels and providing a framework for context. However there remain a number of non-trivial issues to be resolved.

FOHM allows the definition of structure in the form of associations across objects (other associations or data). It controls the access to those associations via the feature space and the accompanying feature vectors. A client that does not understand the feature space should not edit the association as they do not understand the implications of their changes or additions. However FOHM has no easy way of controlling macro-structures formed out of associations, structures formed in the hyperweb itself.

A taxonomic hierarchy is an example of such a structure, one which has strong rules about circularity and branching that cannot be conveyed in the FOHM model (instead they must be enforced at the server end). It is not yet known whether this will be a serious problem to applications that use specific macro-structures.

There also remains a great deal of work to be done on standard feature spaces. An initial set of feature spaces with arrival and traversal semantics is needed. In addition although FOHM represents a powerful framework for the implementation of context we have yet to define a universal context object (either for a user or for data).

The development of FOHM is a step towards a future where structure is as important as data and can be discussed by a variety of different components over a global information network. The authors believe that the infrastructure for that discussion is only now being developed [9] [14] and that FOHM represents the semantic layer that could run over such an infrastructure. One which brings together and extends the traditional information domains.

References

1. 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.
2. Berners-Lee, T., Cailliau, R., Luotonen, A., Nielsen, H. F., and Secret, A. The World Wide Web. *Communications of the ACM 37*, 8 (1994), 76–82.

3. 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.
4. Grønbæk, K. Interoperability - issues beyond the protocol. In *Proceedings of the 4th Workshop on Open Hypermedia Systems, ACM Hypertext '98 Conference, Pittsburgh, PA, June 20-24* (1998), pp. 33–38.
5. Grønbæk, K., and Trigg, R. H. Design issues for a dexter-based hypermedia system. *Communications of the ACM 37*, 3 (Feb. 1994), 40–49.
6. Marshall, C. C., and Shipman, F. M. Spatial hypertext: Designing for change. *Communications of the ACM 38* (1995), 88–97.
7. Marshall, C. C., and Shipman, F. M. Spatial hypertext and the practice of information triage. In *Proceedings of the '97 ACM Conference on Hypertext, April 6-11, 1997, Southampton, UK* (1997), pp. 124–133.
8. 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).
9. 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).
10. Nürnberg, P. J., Leggett, J. J., and Schneider, E. R. As we should have thought. In *Proceedings of the '97 ACM Conference on Hypertext, April 6-11, 1997, Southampton, UK* (1997), pp. 96–101.
11. 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).
12. 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.
13. 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).
14. Waldo, J. Jini architecture overview. Tech. rep., Sun Microsystems, Inc., 901 San Antionio Road, Palo Alto, CA 94303 U.S.A., 1998.
15. 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, Nov 30-Dec 4, Milan, Italy* (1992), pp. 251–261.

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