Exploring the Relationship Between FOHM and RDF

Nicholas Gibbins, Steve Harris, Danius T. Michaelides,
David E. Millard, Mark J. Weal

July 28, 2003

Abstract

One of the results of the Open Hypermedia work of the last decade was the Fundamental Open Hypermedia Model (FOHM) capable of representing contextual structure from a variety of hypermedia domains. As the Semantic Web becomes more important it is interesting to compare its metadata language, the Resource Description Framework (RDF), with a model such as FOHM.

In this paper we examine the relationship between FOHM and RDF to see if they have equivalent expressivity. In particular we look at whether you can represent FOHM structures in RDF, either as a direct mapping from FOHM to the RDF vocabulary or by defining a FOHM schema in RDFS or OWL.

1 Introduction

The Open Hypermedia (OH) community previously tackled the issue of interoperability between different OH Systems with the development of the Open Hypermedia Protocol (OHP) [4]. The intention was to create a reference model and architecture for Open Hypermedia Systems in general.

This change has focused on the fundamental structures that such systems deal with, promoting structures such as associations, references etc. to a first-class status, and the consideration of how context might affect these structures. The Fundamental Open Hypermedia Model (FOHM) [10] has considered how context might be included in the model and specifically how context, and behaviour, can support different views of the hypermedia structures. FOHM also supports and provides interoperability between the different hypermedia domains (Navigational, Taxonomic, Spatial) although this support has been implicit, rather than explicit within the model.

In parallel with developments on OH Systems has been the arrival of the Semantic Web, part of which is the Resource Description Framework (RDF) [7], a language for representing information about uniquely named resources. Designed to function as a way for describing resources on the World Wide Web, RDF can be used to describe any resource, including abstract concepts and objects that have no virtual presence.

RDF and FOHM are both sufficiently expressive enough that it would be possible to describe any FOHM structure in RDF and any RDF graph as FOHM Associations. In this paper we contrast the two models and investigate the implications of implementing the FOHM model using RDF, both in terms of the sacrifices that have to be made and the possible benefits that come from this combination.

Through this comparison and integration, we discuss the strengths and weaknesses of the two representations. Furthermore, since RDF is being promoted as a core standard for the Semantic Web we also discuss the consequences of using RDF to represent complex hypertext structures looking at other developments in this area such as the RDF version of the XLink protocol and systems such as Annotea.

In the following sections we will first provide brief overviews of FOHM and RDF. We then look at whether FOHM could be implemented using just the core RDF vocabulary and the difficulties inherent in this. The next section details a full implementation of FOHM in RDF and OWL, focussing on where the implementation deviates from the core vocabulary, contravening the underlying model theory, and the benefits gained from this approach for FOHM in terms of increased specificity of the hypertext domains. Finally we discuss related work in this area and conclusions that we have drawn from the comparison.

2 RDF

The Resource Description Framework (RDF) [7], is a language for representing information about uniquely named resources. In particular it is designed to function as a way for describing resources on the World Wide Web (for example, authoring information about web pages). However, RDF can be used to describe any resource, including abstract concepts and objects that have no virtual presence.

The Semantic Web initiative is led by the W3C and involves a wide range of institutions and researchers, its purpose is to share information, via the Web, but in a machine parsable format. RDF is the definition language used for metadata on the Semantic Web.

2.1 Describing the Model

RDF relies on identifying resources using Unique Resource Identifiers (URIs). Coupled with a defined set of literal types from XML Schema these can then be used to create a graph of statements, essentially a collection of triples with the form *subject, predicate, object*.

As the predicates are also URIs this graph becomes an unambiguous way for applications to exchange metadata.

There are several notations for RDF triples, graphical and textual, but RDF/XML is the normative syntax that is used. It is a standard way of representing the RDF graph as an XML tree. For brevity, triples can be grouped together in RDF/XML according to their subject, but conceptually they still form a graph.

Because URIs can be very long, RDF/XML uses the XML namespace mechanism to provide a means to abbreviate those URIs used to domain-specific vocabulary (predicates or object types) in an RDF graph. These URIs may then be referred to in the definition using a convenient shorthand (typically of the form *namespace:name*)

Literals may be used in RDF to describe absolute values. Unless otherwise stated these are strings but literals may also be typed. Types can also be defined for resources using RDF Schema (RDFS), which is an RDF-based language for defining domain-specific vocabularies, or *ontologies*. RDFS can be used to define classes, their taxonomic orderings, and the properties which relate instances of classes together. In RDF it is then possible to refer to these classes using the *rdf:type* predicate (which is part of the defined RDF namespace). RDF/XML provides further shorthand to do this without declaring the triple explicitly.

Because the information modelled is often very complex, RDF provides a mechanism for declaring blank nodes, these can be useful for declaring aggregates (for example, if we want to refer to telephone number as country code, area code and number then we need to define these as three separate triples with the same blank node as the subject, any triple wishing to refer to the aggregate telephone number can use the blank node). In RDF/XML blank nodes are given local names to allow the written triples to refer to them but these names are not considered part of the actual conceptual graph.

2.2 RDF Containers

The basic RDF vocabulary contains support for expressing containers (as a particular kind of aggregate). The RDF namespace contains three types of container. *rdf:Bag* (collections where ordering is not important), *rdf:Seq* (collections where ordering is important) and *rdf:Alt* (a collection where each member describes an alternative version of one value or concept).

A container is a resource which has been assigned one of the types above using the *rdf:type* predicate. The members of the container are denoted using the ordinal predicates, which take the form *rdf:_n* where n is a positive integer which represents the position of the member within the container.

2.3 RDF Reification

Using the capabilities of RDF discussed so far, it is not possible to make statements about statements because RDF statements are not first class objects with identity which may be used as the subjects or objects of other statements. RDF contains a *reification* mechanism that makes it possible to elevate a statement to the status of a first class object by creating a resource which represents that statement. It is then possible to make statements about that resource, which are, by proxy, assumed to be about the statement.

A reified statement is represented by a resource of type *rdf:Statement* which is the subject of three statements that represent the relationship between the reified statement and its subject, predicate and object.

Actual "provenance" information is not expressed by the reification vocabulary (i. e. it is impossible to differentiate between different triples with the same subject, predicate and object) but this can be defined at a higher level in an application dependant manner.

2.4 RDF Example

Figure 1 shows an RDF description of this paper. The oval labelled "URI" represents this paper, and has typed arcs which represent its relationship with other entities. It is of type Document (according to the ontology denoted by the prefix "ont"), and has a title and date expressed using the "dc" ontology (Dublin Core, a common metadata vocabulary [2]). It also has a creator, again specified using the Dublin Core vocabulary, which is a sequence of objects (using the rdf:Seq class to maintain the ordering of authors), each of which is given a name using the FN (full name) property from the vCard personal contact information vocabulary (abbreviated as "vc").

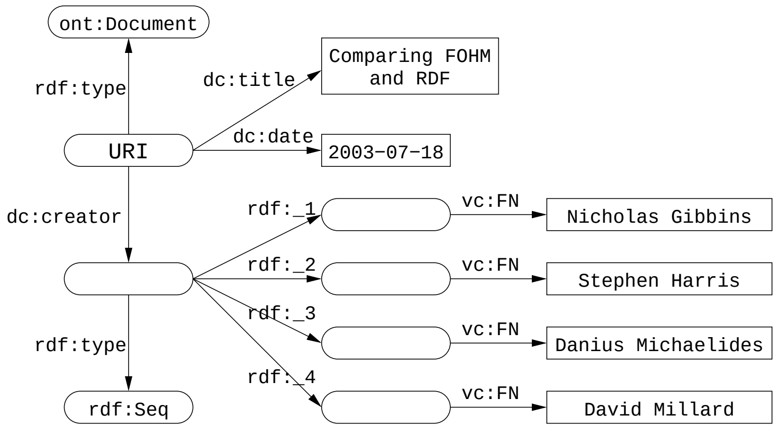

Figure 1: An RDF Graph Describing this Paper

3 FOHM

FOHM is an expansion on the hypertext model defined in the Open Hypermedia Protocol (OHP) [4]. This protocol was developed by the Open Hypermedia Systems Working Group (OHSWG) in an attempt to define an interoperability protocol for Open Hypermedia Systems. OHP divided the various domains of hypermedia (Navigational, Spatial and Taxonomic) into separate protocols (OHP-Nav, OHP-Space and OHP-Tax respectively) each of which had a separate model and series of operations. Only OHP-Nav was defined completely.

Researchers at Southampton believed that it would be possible to expand on the OHP-Nav model so that it was *at least* capable of expressing all three domains [9]. This had several advantages:

Multi-Domain Structures become possible, such as a Navigational Link to a Taxonomic Category).

Cross-Domain Fertilisation occurs, where attributes of one domain become applicable to all, for example Taxonomic branching becomes available with Spatial structures.

An Extendable Set of Structures means that it possible to easily experiment with new domains or variants of the structures in old domains (for example, adding Tours to Navigational Hypertext).

This work resulted in the Fundamental Open Hypermedia Model (FOHM) [10] a generalised model of hypermedia capable of handling the three domains.

3.1 Describing the Model

FOHM is based around two main objects. Data, which are wrappers for resources that are external to the model (either a URL or actual content) and Associations, which are typed collections of either Data or Associations.

As with Dexter [5], FOHM has an anchoring interface that allows the model to reference things that are within Data objects (for example, paragraphs of a text document, scenes of a film). This interface is enshrined in a Reference object which may point to either an Association, a Data, or any part within.

References are collected together in the Associations. Each type of Association defines a feature space and every Reference that is placed within the Association must define its position within that space using a feature vector. For example, a Navigational link has a feature space with a single feature called 'direction', every Reference that is bound to the link must define its direction (either source, destination or bi-directional). The values for each feature are wrapped together within a Binding object.

Thus each Association is a typed collection of Bindings. Each Binding defines the position of a Reference within the Association's feature space. Each Reference may refer, either wholly or in part, to a Data or another Association.

3.2 Context and Behaviour

FOHM defines two types of metadata that may be attached to the model at any point (attached to either Association, Data, Reference or Binding).

The first is totally opaque and is called Behaviour. Behaviour objects wrap metadata that is designed to be application dependant, i. e. it is to be interpreted by the applications processing the FOHM structure. Each Behaviour object has an event tag that describes in which circumstances the metadata is to be applied.

For example, a Link may have a Behaviour attached with the event 'Follow Link'. When certain browsers navigate the structure they might read the metadata and take appropriate action. Examples might be to open the destination in a new browser, to open all destinations simultaneously, etc.

OHP does not include a notion of Behaviour because an event model could not be agreed. FOHM implements Behaviour but at the expense of behavioural interoperability.

FOHM also allows metadata to be attached to model in the form of Context. Context metadata describes in which context this part of the model can be seen, it is expected that processes will use this information to create views on the structure. For example, a Reference that is described as applicable to adults will disappear when viewed by a child. FOHM also describes rules about how structures adjust when objects are removed (the whole branch of the structure is removed up to the parent Association).

Auld Linky [8] is a FOHM-based linkserver that implements this context culling behaviour. FOHM contains the context mechanism in order to implement Taxonomic Hypertext, where Categories must be able to branch according to the Perspective (context) of the viewer. However, Auld Linky has been used with many other structures and the context mechanism has proved particularly useful as a way of implementing Adaptive Hypermedia [1]

3.3 Example FOHM Structure

Figure 2 shows an example FOHM structure. This is a link with one source (the location of the words 'Auld Linky' within a url) and two destinations, the first a short piece of content (plain text), the second another url.

Both destinations explain the name 'Auld Linky', the first concisely, the second with more technical depth. This is described by Context attached to each Binding. If the structure was loaded into Auld Linky and queried using this context then Auld Linky would remove the inappropriate destination.

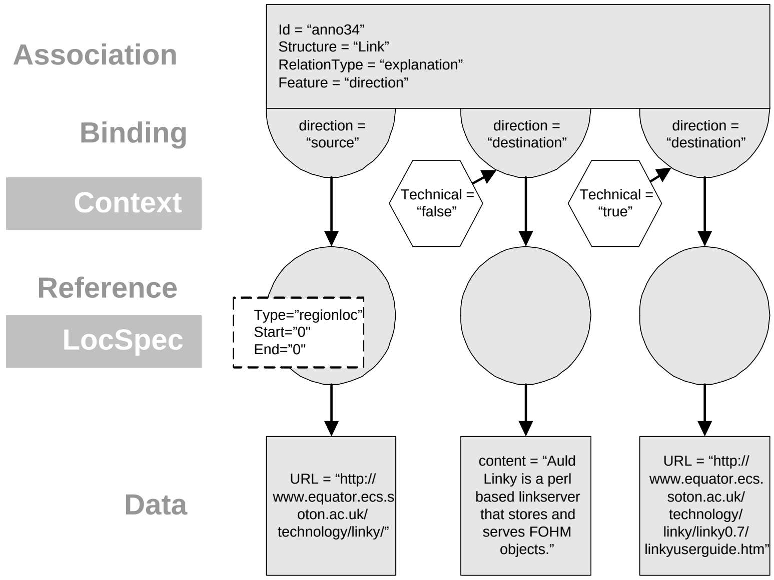

Figure 2: A Simple FOHM Navigational Link

4 FOHM and the RDF Vocabulary

FOHM and RDF are both models for expressing metadata, FOHM is concerned with higher level generic structures (typed n-ary Associations) while RDF operates at a lower level (subject, predicate, object triples). By attempting to implement FOHM using the basic RDF vocabulary we can begin to understand the distance between the two. This allows us to place the work of the Open Hypermedia Community within a Semantic Web context.

The key differences between FOHM and RDF are born out in the first class identity of relations, the complexity of membership to those relations and the way in which content is referred to within the models.

Associations as Nodes. FOHM models its relationships as Association objects but these do not map well into RDF Statements. FOHM Associations have metadata attached (such as type and context information) and are n-ary relations. To model this in the RDF vocabulary it is necessary to translate the Association into an RDF node and each member and metadata attribute into statements about that node.

Reification of Membership Statements. In its container model, RDF allows only unordered or one dimensional ordinal numeral membership. FOHM structures may use a higher-order membership featurespace (such as a Matrix with x, y and z positions) or symbolic rather than numbered feature dimensions (such as the direction of a link). These have no direct mapping into the RDF vocabulary. The only way to represent them is by reifying the RDF membership statements so that many further statements can be made about the membership.

Data as Structured Values. FOHM may refer to external content, or include it within instances of the model, using the Data object. RDF refers to external content using URIs in the same way as it refers to any resource, content can be included directly by using literals. In FOHM the Data object acts as an additional level of indirection which can be referenced many times and have metadata attached to it. To allow the same functionality, the RDF vocabulary includes an *rdf:value* predicate with the intended meaning of proxying the value of a literal to a node so that statements can be made about it.

Using this mapping it is possible to consistently represent FOHM structures using only the RDF vocabulary. However, the mapping relies heavily on reification and structured values, it also conflicts with the Container implementation within the RDF vocabulary.

The RDF model theory that allows you to reason about RDF structures does not have explicit support for these RDF features. The RDF specifications state that reification and structured values have an *intended* interpretation above and beyond that required by the model theory; any understanding of them is application dependant and consequently may not be supported consistently by RDF processors.

For example, a FOHM processor that depends on the understanding that the RDF reification mechanism is to be used to provide the means to express FOHM Association membership cannot rely on other processors to interpret these statements in a similar manner.

In other words, in implementing FOHM in the RDF vocabulary we would lose the advantage of using RDF, as the model theory can no longer be applied. At the same time we are unable to leverage the FOHM semantics because applications would be unable to identify the RDF graph as a FOHM structure.

For this reason, even though both models are generic, it makes sense to define a FOHM Schema in RDFS or OWL to capture the semantics of FOHM while retaining the benefits of RDF.

5 FOHM/OWL

The characterisation of FOHM as an OWL/RDF ontology (FOHM/OWL) has two types of component: the domain-neutral core ontology, which defines the basic notions of associations, bindings, features and so on, and the domain-specific ontologies which describe the different hypermedia domains (navigational, spatial, etc).

This division between core and domain ontologies has been made in order to maximise the interoperability of FOHM structures between clients that do not have a common domain ontology which they both understand. In this case, the clients have a basic understanding of the FOHM structures at the level of the FOHM/OWL core ontology and do not have that deeper understanding of the structures which would be possible if they understood the relevant domain-specific ontology.

The structure of the FOHM/OWL core ontology (see Appendix A) follows the class model of the Java implementation closely. However, it explicitly expresses the relationships between the different classes of FOHM object in a declarative manner, rather than by implicitly embedding their meanings in the code that forms the implementation. Consequently, the ontology makes no assumptions about the actual implementation details.

One notable omission from the core ontology is the type constraint that all the bindings in any given association must contain the same type (or subtype) of feature. This is caused by limitations in the OWL language, which cannot express such general instance-based constraints. However, it is possible to express this in the more specific case found in the domain ontologies, and so the constraint follows from the definition of the domain-specific association types described below.

5.1 FOHM/OWL Domains

RDFS Schemas are analogous to the FOHM Hypertext Domains. In that both define a set of types (and their legal values) and the relationships between them. However, RDFS is a well defined standard while the FOHM Hypertext domains are all implicitly derived from the OHP work.

As explained in the previous section, the FOHM/OWL core ontology is deliberately sparse in order to provide a basic level of interoperability, and the domain ontologies are responsible for describing the structures and constraints that exist in particular hypertext domains.

In FOHM/XML, the type of an association is indicated by the structure and relation attributes of the association object and the contents of the feature element (with the values for that feature on individual bindings being indicated by the featurevalue element). In FOHM/OWL, we create new subclasses of fohm:Association which correspond to these association types, and define them using the OWL vocabulary such that they must only contain bindings which bear the appropriate features (see Appendix B). This is supported by the definition of classes of domain-specific features and the properties which relate them to the new association types.

6 Related Work

6.1 XLink and RDF

FOHM isn't the only model that specifies link structures. The XML Linking specification , better known as XLink, defines ways for XML documents to establish hyperlinks between resources.

Both XLink and RDF provide ways of asserting relations between resources. RDF is primarily for describing resources and their relations, while XLink is primarily for specifying and traversing hyperlinks between resources. A mapping from XLinks to statements in an RDF Model has been defined in a WC3 note [3]. The mapping allows XML Linking elements to be harvested as a source of RDF statements. The idea being that XLinks can provide an alternative syntax for RDF which, in certain situations, may be more useful. The aim is that the harvested RDF models, in some sense, represents the intent of the XML document.

The general principle behind the mapping is that each arc in an XLink gives rise to one RDF statement. The starting Resource of the arc is mapped to the subject of the RDF statement, the ending is mapped to the object and the arc role is mapped to the predicate of the RDF statement. The number of RDF statements harvested from a single arc element will be equal to the number of possible traversals specified by the arc element, with each RDF statement corresponding to one and only one of the traversals.

One problem that arises with the harvesting comes from the fact that in XLink, the linking element itself often serves as one of the participating resources in the link. To model this correctly in RDF, we must be able to define URI references that identify those linking elements. This requires XPointers to be synthesised for these elements.

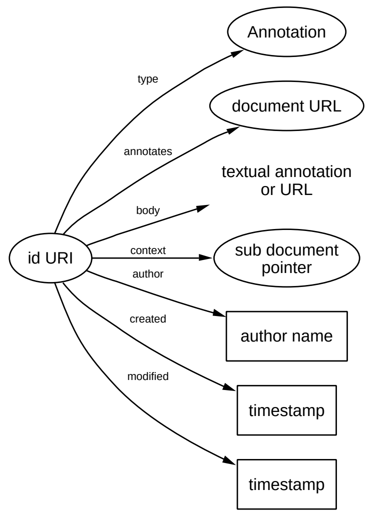

Figure 3: The Form of an Instance of the Annotea Annotation class

An XSLT stylesheet exists that harvests XLinks and generates the appropriate RDF statements.

6.2 Annotea

Annotea [6] is an RDF schema for describing annotations on documents, it relates documents to annotations. The only relations defined in the schema are *body* and *related*, but additional relation types might be added. The *body* property associates a piece of text with the annotation and *related* is an abstract super-property that is intended to be subclassed to produce additional concrete properties.

The document that is annotated is referred to with the *annotates* property and the section of the document to which the annotation applies is indicated with the *context* property. The form of the context is not specified, but typically for XML documents it is an XPointer expression that indicates a subset of the document.

The annotation class also has a number of metadata properties, such as author, name and creation date, modelled after (and sub-classed from) the Dublin Code element set.

7 Conclusions

This paper has set out to examine the relationship between RDF and FOHM, which are both metadata languages capable of expressing statements about data (where data may be other statements). More specifically we looked at whether you can represent FOHM structures in RDF. We believe that this is in fact two questions, firstly whether it is possible to model FOHM structures in basic RDF (i. e. the RDF vocabulary), which would imply an equivalence in their expressivity. Secondly, is it possible to define FOHM using an RDF schema language (RDFS or OWL) and what might be the benefits of doing this.

We have discovered that although it is possible to represent FOHM higher-level relations in the RDF vocabulary, doing so looses the FOHM semantics. In addition the

RDF constructs that are needed to define the FOHM relations break the RDF model theory. This implies that they do not have an equivalent expressivity.

To use the two models together it is necessary to extend the RDF semantics with the higher level FOHM semantics by defining an RDF Schema (in RDFS or OWL). This gives a sound basis to the FOHM structures (unambiguous RDF triples) and preserves the benefits of both RDF and FOHM:

- Consistent way of representing high level relations
- Availability of RDF tools (parsers, validators, processors, etc)
- Possible to use RDFS/OWL to define the FOHM hypertext domains and extend those definitions to explore new domains
- Formal extensible ontology of structure and relation types is possible (e. g. could use a schema to define valid navigational link types for argumentative hypertext)
- Can attach arbitrary metadata with unambiguous predicates

We have also presented our ongoing work to create a FOHM/RDFS schema and OWL domain definitions. We identify the following as further work:

- Reconciling the Linky and RDF query systems
- Defining the original 3 hypertext domains (and explore their overlapping functionality)
- Defining simple Locspecs, Context and Behaviour schema/ontologies
- Assess the performance impact of using an RDF representation of FOHM structures (e. g. by storing them and retrieving them from a triple store [])

This paper is an attempt to bring work from the OHS community into a Semantic Web context by looking at the relationship between FOHM and RDF. It is our hope that this work will help hypertext researchers working with Semantic Web technology to better understand the relationship between RDF statements and Open Hypermedia structures.

8 Acknowledgments

This research is funded in part by EPSRC IRC project “EQUATOR” GR/N15986/01 and EPSRC IRC project ”AKT” GR/N15764/01.

References

[1] Christopher Bailey, Wendy Hall, David E. Millard, and Mark J. Weal. Towards Open Adaptive Hypermedia. In *Proceedings of the International Conference on Adaptive Hypermedia and Adaptive Web Based Systems, Malaga, Spain.*, May 2002.

[2] D. Beckett, E. Miller, and D. Brickley. Expressing Simple Dublin Core in RDF/XML. Technical report, World-wide Web Consortium (W3C) Working Draft (http://www.w3.org/TR/rdf-concepts/), 2003.

[3] Ron Daniel Jr. Harvesting rdf statements from xlinks. http://www.w3.org/TR/xlink2rdf/, June 2000.

[4] Hugh Davis, Siegfried Reich, and David Millard. A Proposal for a Common Navigational Hypertext Protocol. Technical report, Dept. of Electronics and Computer Science, 1997. Presented at 3.5 Open Hypermedia System Working Group Meeting. Aarhus University, Denmark. September 8-11.

[5] Frank Halasz and Mayer Schwartz. The Dexter Hypertext Reference Model. *Communications of the ACM*, 37(2):30–39, 1994.

[6] J. Kahan, M. Koivunen, E. Prud'Hommeaux, and R.R. Swick. Annotea: An Open RDF Infrastructure for Shared Web Annotations. In *Proceedings of the WWW10 International Conference, Hong Kong, May 2001*, May 2001.

[7] G. Klyne and J.J. Carroll. Resource Description Framework (RDF):Concepts and Abstract Syntax. Technical report, DCMI Recommendation, http://dublincore.org/documents/2002/07/31/dcmes-xml/, 2002.

[8] Danius T. Michaelides, David E. Millard, Mark J. Weal, and David C. De Roure. Auld leaky: A contextual open hypermedia link server. In Siegfried Reich and Kenneth M. Anderson, editors, *OHS7 and SC3, Proceedings of the ..., To be published in Lecture Notes in Computer Science, Springer Verlag, Heidelberg (forthcoming)*, 2001.

[9] David Millard, Hugh Davis, and Luc Moreau. Standardizing Hypertext: Where Next for OHP? In Siegfried Reich and Kenneth M. Anderson, editors, *OHS6 and SC2, Proceedings of the ..., Published in Lecture Notes in Computer Science, (LNCS 1903), Springer Verlag, Heidelberg (ISSN 0302-9743)*, pages 3–12.

[10] David E. Millard, Luc Moreau, Hugh C. Davis, and Siegfried Reich. 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 Antonio, TX*, pages 93–102, 2000.

A FOHM/OWL definition (n3 format)

```
<fohm:ReferenceableObject> a <rdfs:Class> ;
  <rdfs:subClassOf ContextualObject> .

<fohm:ContextualObject> a <rdfs:Class> ;
  <rdfs:subClassOf> [ a <owl:Restriction> ;
                      <owl:onProperty> <fohm:hasContext> ;
                      <owl:maxCardinality> "1" ] .

<fohm:Data> a <rdfs:Class> ;
  <rdfs:subClassOf> <fohm:ReferenceableObject> .

<fohm:Association> a <rdfs:Class> ;
  <rdfs:subClassOf> <fohm:ReferenceableObject> .

<fohm:Feature> a <rdfs:Class> .
```

```
<fohm:hasFeature> a <rdf:Property> ;
  <rdfs:domain> <fohm:Association> ;
  <rdfs:range> <fohm:Feature> .

<fohm:Binding> a <rdfs:Class> ;
  <rdfs:subClassOf> <fohm:ContextualObject> ;
  <rdfs:subClassOf> [ a <owl:Restriction> ;
                      <owl:onProperty> <fohm:hasReferenceBound> ;
                      <owl:maxCardinality> "1" ] .

<fohm:hasBinding> a <rdf:Property> ;
  <rdfs:domain> <fohm:Association> ;
  <rdfs:range> <fohm:Binding> .

<fohm:hasRelationType> a <rdf:Property> ;
  <rdfs:domain> <fohm:Association> .

<fohm:hasDescription> a <rdf:Property> ;
  <rdfs:domain> <fohm:Association> .

<fohm:hasReferenceBound> a <rdf:Property> ;
  <rdfs:domain> <fohm:Binding> ;
  <rdfs:range> <fohm:Reference> .

<fohm:Reference> a <rdfs:Class> ;
  <rdfs:subClassOf> <fohm:ContextualObject> ;
  <rdfs:subClassOf> [ a <owl:Restriction> ;
                      <owl:onProperty> <fohm:hasReferencedObject> ;
      <owl:maxCardinality> "1" ] .

<fohm:hasReferencedObject> a <rdf:Property> ;
  <rdfs:domain> <fohm:Reference> ;
  <rdfs:range> <fohm:ReferenceableObject> .

<fohm:hasLocSpec> a <rdf:Property> ;
  <rdfs:domain> <fohm:Reference> ;
  <rdfs:range> <fohm:LocSpec> .

<fohm:LocSpec> a <rdfs:Class> .

<fohm:hasContext> a <rdf:Property> ;
  <rdfs:domain> <fohm:ContextualObject> ;
  <rdfs:range> <fohm:Context> .

<fohm:Context> a <rdfs:Class> .

<fohm:hasBehaviour> a <rdf:Property> ;
  <rdfs:domain> <fohm:ContextualObject> ;
  <rdfs:range> <fohm:Behaviour> .

<fohm:Behaviour> a <rdfs:Class> .
```

B OWL Domain Definition: Navigational Hypertext (n3 format)

```
<nav:DirectionAssociation> a <rdfs:Class> ;
  <rdfs:subClassOf> <fohm:Assocation> ;
  <rdfs:subClassOf>
    [ a <owl:Restriction> ;
      <owl:onProperty> <fohm:hasBinding> ;
      <owl:allValuesFrom>
        [ a <owl:Class> ;
  <rdfs:subClassOf> <fohm:Binding> ;
          <rdfs:subClassOf>
            [ a <owl:Restriction>
      <owl:onProperty> <fohm:hasDirectionFeature> ;
              <owl:cardinality> "1" ] ] ] .

<nav:hasDirectionFeature> a <rdf:Property> ;
  <rdfs:subPropertyOf> <fohm:hasFeature> ;
  <rdfs:domain> <nav:DirectionAssociation> ;
  <rdfs:range> <nav:DirectionFeature> .

<nav:DirectionFeature> a <rdfs:Class> ;
  <rdfs:subClassOf> <fohm:Feature> .
```