From Metadata to Links

Graham Moore[1,2] and Luc Moreau[2]

[1] STEP UK, gdm@stepuk.com
[2] Department of Electronics and Computer Science, University of Southampton, L.Moreau@ecs.soton.ac.uk

Abstract. Metadata systems are considered to be a powerful and generalised mechanism for extending the properties that constitute an object and for facilitating access to information. However, there is another mechanism, linking, that can also be considered as having the ability to extend the properties of an object and provide systems and users with more useful means of accessing information. In this paper, we represent both metadata and linking as abstract models and show how the metadata model may be expressed in terms of the linking mechanism. Then we discuss the benefits of the more general model of linking.

1 Introduction

Metadata systems exist for two primary reasons. Firstly, they provide a generic mechanism for the extension of an object within a system. This object need not be confined to be a document, as is classically associated with metadata systems, but should be considered to be any object [2]. The metadata mechanism works by associating a name and value pair with an object. This provides the ability to extend the attributes of an object without needing to change the object itself.

Secondly, metadata is used for grouping, sorting and providing access to objects in a system. In this scenario the metadata system can be used to provide an "Information Retrieval" service. For example, it would be possible to retrieve all objects that have a metadata value called 'date' where the value of 'date' is before '240300'.

Linking also provides us with the ability to associate objects and convey the semantics of why we associate particular objects together. Using an open-hypertext linking model [8] we can encapsulate the association information independently of the associated objects. We can search and query those data structures similarly as we do with metadata.

There are obvious similarities between links and metadata. The purpose of this paper is to investigate such similarities and better our understanding of these two models. Our first thesis is that open-linking models can express metadata functionalities and more.

Unlike the metadata model the linking model is symmetric. This is because all anchors in a link have similar "weights" and are each identified by their role. In a link, depending on a perspective[1], any anchor can be regarded as associated

[1] Note that we do *not* use the technical meaning of perspective [3].

with any other anchor. Such an association indicates the presence of information related to an object, which is precisely the idea conveyed by metadata. Thus, our second thesis is that metadata can be expressed as a perspective on typed links.

2 From Metadata to Linking

The power of an information retrieval system can be measured by how versatile the retrieval mechanism is and how explicitly it conveys the intended semantics of the user or process that built the system. In this section we present our perception of the fundamental properties of both metadata and linking systems and express them in terms of a formal model. By doing so we can decipher the power of each system and ultimately express one in terms of the other.

In this paper we perform five steps to express a metadata model in terms of an open linking model. The five steps are:

1. We define a model of metadata.
2. We summarise the simple linking model formalised in [5].
3. We show how to define the metadata model in terms of the simple linking approach. This is an enabling step towards a richer structure but we also discuss the properties of this definition.
4. We generalise the simple linking model into a notion of typed links, similar to the OHP-Nav notion of link [8].
5. We define the metadata model in terms of typed links.

The latter definition preserves the original abilities of the metadata model yet extends the power through the use of additional properties, such as role identification and association typing that are found in the linking model.

2.1 A Model for Metadata

In this section we define metadata systems, or the role of metadata, allowing:

1. New information to be associated with an object without the need to edit that object in any way.
2. Objects to be managed (grouped, sorted, navigated to) by the content of the associated metadata.

The *traditional view* of metadata is in the form of a pair, composed of an attribute and a literal value. This pair alone is inadequate to describe a metadata association. The model is completed with the introduction of the object with which the metadata pair is being associated.

In order to identify and manipulate the different components involved in a metadata system, we formalise its data model. In Figure 1, we consider a finite set of objects, where each object has the potential to be extended by metadata. Each object is identified by a unique name, belonging to the set of names $Name$. A naming function maintains the mapping between names and objects: in practical terms, given a object name, a naming function ρ is able to return the associated object.

We slightly extend the traditional view by regarding a metadata pair as composed of a metadata field and metadata name, respectively belonging to sets $Field$ and $Name$. The name component of the metadata pair maps to an object. The intended meaning is that the value of this field is this object.

By defining the name aspect in a metadata pair, we are encouraging the view that metadata systems do not introduce special kinds of values but utilises the existing named object mechanism. Finally, the intent of a metadata system is to associate metadata pairs with objects. We formalise this notion by introducing a metadata association, composed of an object name and a metadata pair.

$$
\begin{array}{rlr}
N \in Name &= \{N_1, N_2, \ldots\} & (Object\ Name) \\
o \in Object &= \{O_1, O_2, \ldots\} & (Object) \\
\rho \in Env &= Name \rightarrow_f Object & (Naming\ Function) \\
\\
F \in Field &= \{F_1, F_2, \ldots\} & (Metadata\ Field) \\
p \in Pair &= Field \times Name & (Metadata\ Pair) \\
assoc \in Assoc &= Name \times Pair & (Metadata\ Association)
\end{array}
$$

Notation:

$$
\begin{aligned}
assoc &:= \langle N, p \rangle \\
p &:= \langle F, N \rangle
\end{aligned}
$$

Fig. 1. The metadata model

Now follows an example based on the abstract model defined in Figure 1. Using this example we then discuss the problems with this metadata model.

Example 1. In the following metadata representation,

$$\langle \texttt{"doc1"}, \langle \texttt{"author"}, \texttt{"graham"} \rangle \rangle,$$

`"author"` is a field, `"graham"` is the name of an object, and `"doc1"` is the name of an object (here representing a document in a document management system).

Information retrieval systems are more powerful the more they capture, express and allow for searching of the semantics that relate information. The more precise the semantics the more accurate grouping and associations can be. We now discuss the semantics expressed by the metadata model.

This metadata model requires that the semantic is implied by the 'reader', where the 'reader' may be a human or computer system. The metadata association could be read as:

"graham" is the author of "doc1" (1)

Thus the human or computer may then express:

"doc1" is the document that was authored by "graham" (2)

Sentence (2) can only be infered through the assumption that the name “doc1” maps to a document. We have defined our metadata model such that metadata can be associated with any object and not just documents. Given this we are unable to infer (2), unless we have some other mechanism for finding out the type of the object.

One such mechanism may be to access the object, but this breaks the metadata model where metadata is expressed and encapsulated independently of the object. In this case we do not have an encapsulation of the association semantics as we need to ’touch’ the object in order to complete parts of the metadata structure.

In the interpretation we can see two other pieces of information that a system could query, sort on or display to the user. The two pieces of information provided are ’authored by’ and ’document’, respectively, the reason for the association of these objects and the role of one of the objects within the association. This infered information is not part of the metadata model, making it explicit would allows us to access and process it.□

While metadata initially appears to be a generic model for defining data associations with meaning, it transpires that the actual semantics of the metadata model are weak. Ultimately we have to question the view of metadata as a generalised association mechanism, for we have seen how assumptions lead to particular interpretation of the semantics.

2.2 Model for Simple Links

In this section we introduce the Simple Linking Model. This model will be used when we define the metadata model in terms of Linking.

In Figure 2, we present the key aspects of the simple link model, as defined in [5]. We assume that objects are identified by names, and that there is a mapping function, associating each name with an object. A link is a first-class entity, separate from objects, which we define as a pair of anchors. Anchors can be considered as handles onto the objects being associated by the link, thus they are simply defined in terms of a name.

$$
\begin{array}{rll}
N \in \mathit{Name} = & \{N_1, N_2, \ldots\} & (\mathit{Object\ Name}) \\
o \in \mathit{Object} = & \{O_1, O_2, \ldots\} & (\mathit{Object}) \\
\rho \in \mathit{Env} = & \mathit{Name} \rightarrow_f \mathit{Object} & (\mathit{Naming\ Function}) \\
L \in \mathit{Link} = & \mathit{Anchor} \times \mathit{Anchor} & (\mathit{Link}) \\
A \in \mathit{Anchor} = & \mathit{Name} & (\mathit{Link\ Anchor})
\end{array}
$$

Fig. 2. Simple Linking model

We can see that the association created by a simple link is symmetric; we will use this property in the following section.

2.3 From Metadata to Simple Links

The traditional metadata model is “readable” in one direction only and prohibits a meaningful inverse interpretation. Situations arise where it is unclear which object is the metadata and which is the data; within one association both objects could play both roles. The metadata model is not rich enough to do this but the linking model is.

In this section we proceed with the definition of the metadata model in terms of the simple linking model described in Section 2.2. The motivation for doing this is that we want to make the metadata model symmetric so that:

- The model is generic in that metadata is the combination of objects, symmetric associations and perspective.
- We can further extend it to use typed links, thus explicitly capturing the full semantics of the association.

We consider the following revised definition of the naming function:

$$\begin{aligned} \mathit{Pair} &\subseteq \mathit{Object} \\ \rho \in \mathit{Env} &= \mathit{Name} \rightarrow_f \mathit{Object} \end{aligned}$$

A metadata association *assoc* of the form $\langle N, p \rangle$ can be expressed in the simple linking model by the pair $\langle N, N_p \rangle$ where N_p is a new name such that

$$\rho(N_p) = p.$$

Fig. 3. The Metadata to Linking Transformation

Figure 3 defines how the asymmetric metadata model may be transformed into the symmetric linking model. The key step is to treat the metadata pair as an identifiable object within the system. Thus, metadata pairs are defined as being a subclass of *Object*. This then allows an anchor, via its name, to be mapped using the standard naming function to a metadata pair.

Example 2. If `"doc1"` is the name denoting a document, and `"p"` the name denoting a pair ⟨`"author"`, `"graham"`⟩, then ⟨`"doc1"`, `"p"`⟩, is the representation of the metadata seen in Example 1 in terms of a simple link.□

This definition is a step towards a model of typed links that can explicitly expose the full semantics of an association. However, this intermediate model is worth discussing because of the following properties:

1. Now that the metadata pair is an object, it is necessary to understand the internal structure of this object in order to access and interpret the metadata.

In the same way that the previous example required us to 'touch' the object to find out it was a document this model requires us to fetch and interrogate an object in order to discover the metadata field and object value name.
2. Also similar to the metadata model, the semantics for the association are embedded within an object as opposed to being encapsulated within the association itself.
3. The third issue with this model is that it provides no mechanism for expressing the meaning of the association and of *all* components in the association (e.g, here the role of "document" is not explicit).

This simple linking model does not provide greater expressiveness than the metadata model and additionally requires us to fetch and examine an object. However, moving from an asymmetric to a symmetric model enables us to define the metadata model in terms of typed links, which we do in the following section.

2.4 Model for Typed Links

Having now developed a symmetric model from the asymmetric metadata model we have seen that there is more work to be done. In this section we extend the simple linking model to have typed links. Later we will see that by using this model we can fully define the semantics of the association and the roles of the anchors within it.

The extension of the linking model introduces the ability to specify the semantics that describe the reason for associating objects together and allow for the role within the link of each participating anchor to be defined.

In Figure 4, we generalise the notion of simple links by the notion of typed link. A typed link is a triple formed of a type and two anchors. Each anchor contains the name of an object and its role in the link. For example, a directional link for navigation may give the role of source and destination to the two anchors in the link.

$$
\begin{array}{rll}
ta \in \mathit{TypedLink} = \mathit{Type} \times \mathit{Anchor} \times \mathit{Anchor} & (\mathit{Typed\ Link}) \\
a \in \mathit{Anchor} = \mathit{Name} \times \mathit{Role} & (\mathit{Anchor\ with\ Role\ Type}) \\
r \in \mathit{Role} = \{R_1, R_2, \ldots\} & (\mathit{Role}) \\
t \in \mathit{Type} = \{T_1, T_2, \ldots\} & (\mathit{Link\ Type})
\end{array}
$$

Fig. 4. The Typed Link Model

We have introduced the typed linking model such that we can fully convey the intended semantics that caused these objects to be associated. Now that we have such a mechanism our systems can provide more refined object-retrieval as there is valuable information that can be processed. The next stage is to show how the metadata model can be mapped into this one, thus giving the new metadata system the equivalent power of this typed linking model.

2.5 From Metadata to Typed Links

We have shown how the metadata model is not as general as first thought and that the semantics it communicates are not rich enough. Therefore the value of an information system built on top of such a mechanism is questionable. It should be noted that we do not discard the value of this kind of system, we are merely illustrating that more valuable semantics can be added while providing an equivalent metadata model.

In Figure 5, we construct a typed association from three of the components of a metadata association, namely a name N, a field F and its associated name N'. The transformation requires us to identify two further components: T the type of the association and R^N the role of N in the association of type T. They may be used to construct an instance of typed link, explicitly containing the name N and N' and the field F, which were part of the initial metadata association.

Let us consider a metadata association $assoc$ of the form $\langle N, p\rangle$, with p of the form $\langle F, N'\rangle$.
Let R^N be the role of the object named N in an association of type T.
We can construct an instance of $TypedLink$ representing $assoc$ as follows:

$$\langle\langle N, R^N\rangle,\ \langle N', F\rangle,\ T\rangle.$$

Fig. 5. The Metadata to Typed Link Transformation

The key aspect about the definition in Figure 5 is that the metadata model of Figure 1 is not rich enough to populate the typed link model, yet the typed link is powerful enough to express all the aspects of the metadata model. Given this, information systems could define metadata in terms of linking and then fill in the undefined semantics to provide a richer model. The example below illustrates a metadata to linking transition.

Example 3. Let us reconsider the metadata introduced in Example 1.

$$\langle \texttt{"doc1"}, \langle \texttt{"author"}, \texttt{"graham"}\rangle\rangle$$

We can transform this example into the typed link model as shown below. Note, we have left question marks where the metadata model is insufficient to fully utilise the typed link model.

$$\langle\langle \texttt{"doc1"}, ?\rangle, \langle \texttt{"graham"}, \texttt{"author"}\rangle, ?\rangle.$$

Finally, we instantiate the missing semantics. We now have a model that encompasses what was originally expressed but has additionally provided richer association semantics.

⟨⟨"doc1", "document"⟩, ⟨"graham", "author"⟩, "DocumentAuthoredBy"⟩

Figure 6 shows a graphical representation of the previous link, the roles it contains and its type.□

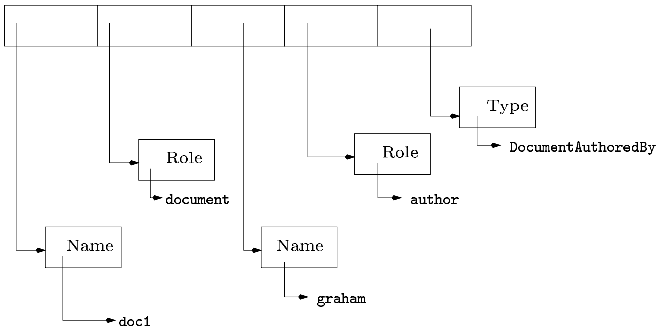

Fig. 6. Metadata as Typed Link

The next example illustrates that due to symmetry of the linking paradigm we can now consider metadata to be typed links and perspective.

Example 4. In the following link example the object whose name is "luc" plays several roles in different associations. Depending on a given perspective the object named "luc" can be considered as metadata to the object named "graham" and the object named "doc2".
⟨⟨"doc2", "document"⟩, ⟨"luc", "author"⟩, "DocumentAuthoredBy"⟩
⟨⟨"graham", "supervisee"⟩, ⟨"luc", "supervisor"⟩, "Supervises"⟩ □

This section has shown how the metadata model can be defined in terms of typed links and that this step leads to a more expressive model. Secondly, we have shown by the use of examples how metadata can be viewed as typed links and perspective.

3 Discussion and Related Work

The Open Hypermedia Systems Working Group (OHSWG) was formed to coordinate the efforts of the community in creating standards that would address interoperability issues. Among the output of this working group, we find the Open Hypermedia Protocol (OHP). In particular, OHP [8] defines a data model, identifying all the data structures necessary to represent information about "hyper links" and associated primitives. Furthermore, OHP aims at providing a medium to promote interoperability between hypertext domains. From the data model specified by OHP, a fundamental open hypertext model (FOHM) [4] was formally defined and showed to support interoperability.

In this paper, we have assumed the existence of a linking data model to express the notion of metadata. We can show that this model is entirely compatible with the FOHM model. Indeed, we observe the following correspondence:

Metadata Linking Model	FOHM
typed links	associations
role	feature
link type	relation type

As a result, we conclude that the OHP and FOHM data model are powerful enough to encode the type of metadata that we have presented in this paper.

Structural computing [6] has emerged as an alternative approach to unify hypertext domains, including domains such as spatial and taxonomic hypertexts. The support of our metadata by structural computing would require the definition of a new "middleware" able to handle metadata appropriately. Lessons from our experiments can benefit the design of such middleware. In particular, symmetry in the link between the metadata and the object it refers to should be preserved in order to allow for powerful queries over metadata.

This paper has highlighted a number of issues that could provide a basis for further work. The introduction of names to replace objects has allowed us to think and model in more generic terms. It would be interesting to investigate the use of named objects as the role and type parts of a typed link. Continuing this theme, a more detailed investigation into the nature of a system where links are members of the set of objects could produce some interesting results. Showing this would allow links to associate links together; could this be used as a formal model for standards such as Topic Maps [1] ?

4 Conclusion

This paper has shown how the metadata model can be defined in terms of both a simple and typed linking model. Defining metadata using simple links did not enrich the model but illustrated that it was possible to make a metadata model symmetric. Knowing this, we defined the metadata model in terms of the typed linking model. This definition showed how the metadata model was less expressive than the linking model, and how the typed linking model was capable of capturing the intent of the original metadata association. Given the linking model was more expressive we showed how the missing additional semantics could be added in order to complete the enrichment of the original metadata association. Our second thesis, that metadata can be expressed as a perspective on typed links, was shown by defining the metadata model in terms of typed links and that examples in this typed link form could be seen as metadata when viewed from a given perspective.

References

1. Michel Biezunski, Martin Bryan, and Steve Newcomb. Topic maps. ISO/IEC FCD 13250:1999.
2. Dublin Core Metadata Initiative. `http://purl.oclc.org/dc/`.
3. Catherine C. Marshall and Frank M. Shipman. Spatial hypertext: Designing for change. *Communications of the ACM*, 38:88–97, 1995.
4. Dave Millard, Luc Moreau, Hugh Davis, and Sigi Reich. FOHM: A Fundamental Open Hypertext Model for Investigating Interoperability between Hypertext Domains. In *Proceedings of the Hypertext Conference HT'00*, 2000.
5. Luc Moreau and Wendy Hall. On the Expressiveness of Links in Hypertext Systems. *The Computer Journal*, 41(7):459–473, 1998.
6. Peter J. Nürnberg, Kaj Grønbæk, Dirk Bucka-Lassen, Claus A. Pedersen, and Olav Reinert. A component-based open hypermedia approach to integrating structure services. *New Review of Hypermedia and Multimedia*, 5:179–207, 1999.
7. Resource Description Framework (RDF). `http://www.w3c.org/RDF/`.
8. Siegfried Reich, Uffe K. Wiil, Peter J. Nürnberg, Hugh C. Davis, Kaj Grønbæk, Kenneth M. Anderson, David E. Millard, and Jörg M. Haake. Addressing interoperability in open hypermedia: The design of the open hypermedia protocol. *New Review of Hypermedia and Multimedia*, 5:207–248, 1999.