Making RDF Presentable:

Selection, Structure and Surfability for the Semantic Web

Lloyd Rutledge, Jacco van Ossenbruggen and Lynda Hardman[*]
CWI
P.O. Box 94079
NL-1090 GB Amsterdam, The Netherlands
email: Firstname.Lastname@cwi.nl

ABSTRACT

This paper discusses generating document structure from annotated media repositories in a domain-independent manner. This approaches the vision of a uniform and universal RDF browser. We start by applying the search-and-browse paradigm established for the WWW to RDF presentation. More importantly, this paper adds to this paradigm the clustering-based derivation of document structure from search returns, providing simple but domain-independent hypermedia generation from RDF stores. While such generated presentations hardly meet the standards of those written by humans, they provide quick access to media repositories when the needed document has not yet been written. The resulting system allows a lay user to specify a topic for which it generates a hypermedia document providing guided navigation through a previously unfamiliar RDF repository. The impact for content providers is that as soon as new media items and their annotations are added to a repository, they become immediately available for automatic integration into subsequently requested presentations.

Categories and Subject Descriptors

H.5.1 [**Information Systems**]: Multimedia Information Systems; I.7.2 [**Computing Methodologies**]: Document and Text Processing—*Document Preparation*; H.5.4 [**Information Systems**]: Information Interfaces and Presentation—*Hypertext/Hypermedia*; I.2.4 [**Computing Methodologies**]: Artificial Intelligence—*Knowledge Representation Formalisms and Methods*

General Terms

Design, Documentation, Human Factors, Standardization

Keywords

Semantic Web, hypermedia generation, media archives, browsing, search, clustering, RDF

[*]Lynda Hardman is also affiliated with the Technical University of Eindhoven.

WWW2005, May 10–14, 2005, Chiba, Japan.
.

1. INTRODUCTION

The explosive adoption of HTML and the WWW is due in large part to its immediate delivery from author to user: once the author encodes a document in HTML and posts it, any user anywhere can access it without needing specialized software. Most assume the Semantic Web can have no such immediate accessibility, being instead accessible only indirectly through user interfaces encoded for specific domains. One key factor in this assumption is that the Semantic Web lacks the document structure HTML and other XML formats have: primarily, that of hierarchy and sequence. Of course, RDF intentionally lacks these, choosing instead to facilitate machine-processing of the data it encodes. However, this focus on machine-process does not necessarily preclude immediate accessibility from humans — it just makes such access more complex. Lacking document structure means lacking the document form all users are familiar with, making many RDF interfaces unapproachable to lay users. Converting RDF structure to XML document structure in a domain-independent manner would give the information it encodes the same accessibility and approachability HTML enjoys. However, the automated generation of sensible, informative document structure from a source without such structure remains a difficult problem, as does domain-independent processing of RDF.

This paper proposes overcoming these obstacles of domain-independence and document structure with a three-pronged approach: using the *media* accessible via RDF, *clustering* RDF components to build document structure and applying observations of the *human processes* of the document author and web user. It is this web user who inspires the three keywords of this paper's subtitle: selection, structure and surfability. The typical web user begins access to the enormous WWW by *selecting* a subset in the form of a search query specifying the topic of interest. That the pages selected have document *structure* makes them readable to the lay user, clarifying the relations between their components. *Surfability*, or browsing, means the user has the freedom to go beyond the computer-generated selection and the human-written structure when either doesn't perfectly suit the user. Just as surfability has enabled lay users to work beyond the inaccuracies of computer-generated selection, this paper argues that it can also let lay user get beyond the similar inaccuracies computer-generated document structure brings, allowing useful presentation generation to occur.

Our goal is to generate navigable structures that orient each user in the current local context, communicates the overall structure from this perspective and enables the user to navigate through it while maintaining a sense of position in the information space. Our key assumption is that the strategies human document authors deploy to convey information to their readers can also apply, to a certain extent, to the automated presentation of Semantic Web data. Using this metaphor helps improve the readability of flat lists of RDF triples by ordering, grouping and prioritizing information before presenting it.

There are many types of RDF use, and while some don't apply to this paper's style of direct presentation, many do. One primary example category is that of repositories of annotated media objects, especially when these objects are not whole documents but are instead small enough to function as components of generated documents. Another applicable category we identify is "conceptual RDF", which defines abstract concepts, relates them to each other and associates them with media for conveying them. We created such a conceptual RDF repository to test this work's premises. This is our conversion to RDF of the ARIA (Amsterdam Rijksmuseum Interactive) database, which drives the interface to its collection website [17]. ARIA defines about 1250 artifacts from the museum, associating them not just with images but also with other concepts such as description, genres, detail and artists.

After a review of related work in Section 2, Section 3 presents the determination of a presentation's document structure. We follow up on this in Section 4 by describing our approach's application to generating the individual displays the user navigates between. Finally, we wrap up with a summary and conclusions.

2. RELATED WORK

While browsing a document repository with a relatively small number of large chunks of information, with few explicit relations among these chunks, a user might succeed without the help of an interface that makes the underlying structure explicit, such as a sitemap or fish-eye view. With RDF, the situation is typically the exact opposite: we have many small chunks of information with many explicit relations among them. The user interfaces of many RDF tools clearly reflect aspects of this observation. This section discusses current approaches to displaying RDF.

2.1 Local Interface

There are several approaches to browsing RDF that are domain-independent. These are all small-scale, working at the level of the triples that RDF defines. Here we discuss the following approaches to domain-independent localized browsing: as code, as navigation and as an editing interface.

RDF as XML. The typical first interface for working with RDF, particularly in the absence of extensive domain-specific processing, is its XML serialization [14], exemplified in Figure 1. While this author-controlled grouping around the subjects in the triples provides some potentially useful structure beyond the unsorted and unstructured triples RDF encodes, it is clearly difficult to read. Serialization syntaxes such as n3 [5] have been proposed as a more human-friendly alternative for the XML-based serialization, but are, of course, also poorly suited for reading and extensive browsing.

Sesame Explore Mode. The explore mode of the Sesame open source RDF database management system provides a more browser-like interface to RDF, as shown in Figure 2 [7]. It shows all RDF triples in which a given URI features in the subject, property or object of the triple. A link from each triple component generates a similar page for that URI, making RDF "surfable". Also note that while many URIs in RDF are only used as an identifier, many others actually resolve into a meaningful resource on the Syntactic Web. This is acknowledged in the interface by the "Visit this URI on the Web" link under the URI. The current view is always limited to the immediate vicinity of the current resource. Additionally, by producing a flat list of triples, the Explorer does not show the underlying structure.

```
<rdf:Description rdf:about='&ARIA;#ArtefactSK-A-2670'>
 <rdf:type rdf:resource='&ARIA;#Artefact'/>
 <dc:title>Pinks in the Breakers</dc:title>
 <aria:thumbnail resource=
   'http://www.rijksmuseum.nl/.../SK-A-2670.org.jpg'/>
 <dc:creator rdf:resource='&ARIA;#Artist11960'/>
 <dc:date>1875-1885</dc:date>
 <dc:description>
  ... along the beach by horses. Scheveningen did not ...
 </dc:description>
 <aria:artefactMaterial>
    Oil on canvas
  </aria:artefactMaterial>
 <aria:artefactSize>90 x 181 cm</aria:artefactSize>
 ...
 </rdf:Description>
```

Figure 1: XML serialization of an RDF encoded Rijksmuseum Artifact

Protege-2000. Semantic Web editors such as Protege-2000 [12] offer additional browsing possibilities. Protege-2000's emphasis is more at the level of RDFS than RDF. It provides an extensive interface for browsing the hierarchies defined by RDFS subclasses. The class instance interface of Protege-2000 also provides is similar to Sesame explore mode's navigation along triples.

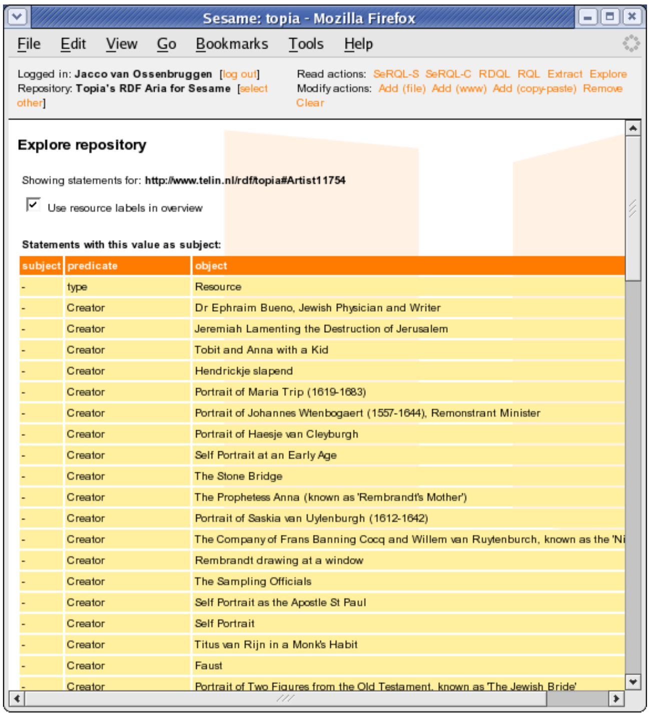
2.2 Global Interface

In contrast to local interfaces, several systems provide domain-independent large-scale views of RDF repositories. Whereas small-scale browsing focuses on content, these large-scale viewers focus on the broad relational structuring joining the content. This is often a trade-off: just as the browsers above fail to give the big picture, the systems we present here typically have poor presentation of the detailed content.

RDF Graph Generation. The most generic, both in terms of visual technique and the domains is applies to, global interface to RDF is probably the W3C's RDF Validator [15]. This system provides a graph-based interface to any RDF repository. Figure 3 shows the hyperlinked SVG version of such a graph. It automatically generates a graph-based view of the validated triples. While this gives a user some information about the underlying structure, in particular with some grouping performed by its layout algorithm, it does little to group, order or prioritize information. Another well-known drawback of this is the limited scalability: with large numbers of triples, the graph quickly becomes unmanageably large.

AutoFocus. An example of a more interactive alternative for navigating structure appears in Figure 4 [3]. This diagram results from running the ARIA RDF store through an adapted version of AutoFocus [1]. Generally speaking, AutoFocus groups resources based on a set of keywords given by the end-user, showing directly what keyword is associated with what resource, and, more importantly, which resources share a common set of keywords. Here, it takes selected resources in ARIA and uses the same visualization to show clusters derived from their common characteristics. The AutoFocus cluster interface renders resources as yellow dots and, except for a few labels, shows no textual content. In contrast, Sesame Explorer and W3C RDF Validator show every URI and every literal that constitute the RDF triples displayed. This not only raises the question how much should be shown in what situation, it also raises the more fundamental question of what precisely is the "content" of a given set of RDF triples.

mSpace. mSpace [11] bring global interfaces closer to our work here by deriving global structure for exploring relational data



Sesame: topia - Mozilla Firefox

Logged in: **Jacco van Ossenbruggen** [log out]
Repository: **Topia's RDF Aria for Sesame** [select other]

Read actions: SeRQL-S SeRQL-C RDQL RQL Extract Explore
Modify actions: Add (file) Add (www) Add (copy-paste) Remove Clear

Explore repository

Showing statements for: **http://www.telin.nl/rdf/topia#Artist11754**

☑ Use resource labels in overview

Statements with this value as subject:

subject	predicate	object
-	type	Resource
-	Creator	Dr Ephraim Bueno, Jewish Physician and Writer
-	Creator	Jeremiah Lamenting the Destruction of Jerusalem
-	Creator	Tobit and Anna with a Kid
-	Creator	Hendrickje slapend
-	Creator	Portrait of Maria Trip (1619-1683)
-	Creator	Portrait of Johannes Wtenbogaert (1557-1644), Remonstrant Minister
-	Creator	Portrait of Haesje van Cleyburgh
-	Creator	Self Portrait at an Early Age
-	Creator	The Stone Bridge
-	Creator	The Prophetess Anna (known as 'Rembrandt's Mother')
-	Creator	Portrait of Saskia van Uylenburgh (1612-1642)
-	Creator	The Company of Frans Banning Cocq and Willem van Ruytenburch, known as the 'Ni
-	Creator	Rembrandt drawing at a window
-	Creator	The Sampling Officials
-	Creator	Self Portrait as the Apostle St Paul
-	Creator	Self Portrait
-	Creator	Titus van Rijn in a Monk's Habit
-	Creator	Faust
-	Creator	Portrait of Two Figures from the Old Testament, known as 'The Jewish Bride'

Figure 2: Display from Sesame's Explore Web interface

stores, including those encoded in RDF. However, its metaphor is a multi-dimensional grid instead of document structure. mSpace's interface is a table whose columns each represent one "dimension", which consists of the different values for a particular property the repository components have. By selecting cells in each column from left to right, the user specifies incremental subsets having those cells' property assignments. Users can change this column order. mSpace's interface resembles more a relational database or spreadsheet than a document. Its focus is at a higher level in the informative structure than what this paper presents, with quicker navigation and dynamic transformation. mSpace's building of orthogonal dimensions relies on relatively uniform property types for the items it provides access to, whereas our approach allows more variation.

2.3 Hybrid Interface

While small-scale browsing is like missing the forest for the trees, the large-scale view is comparable to exploring a forest from an airplane, with no way to land. The two approaches are combined in most traditionally formed documents, with the overall structure woven in with the detailed content. While this interweaving of scales has thus far proven difficult to automate, several systems for automatic generation of hypermedia from meta-data repositories have made some progress.

Haystack. The Haystack framework [16] is, at the time of writing, the most well known approach to the "view RDF as a document" metaphor. Haystack aims at providing a Semantic Web-based personal information management system, integrating (Semantic) Web browsing with email and calendar tools. Haystack features its own RDF manipulation language (Adenine) and a separate RDF presentation language (Ozone). The latter can be used to define style sheets for specific RDF vocabularies or applications. In addition, the fact that Haystack is an end-user client tool combined with the special purpose languages make it harder to realize server-side integration of RDF content with more traditional content.

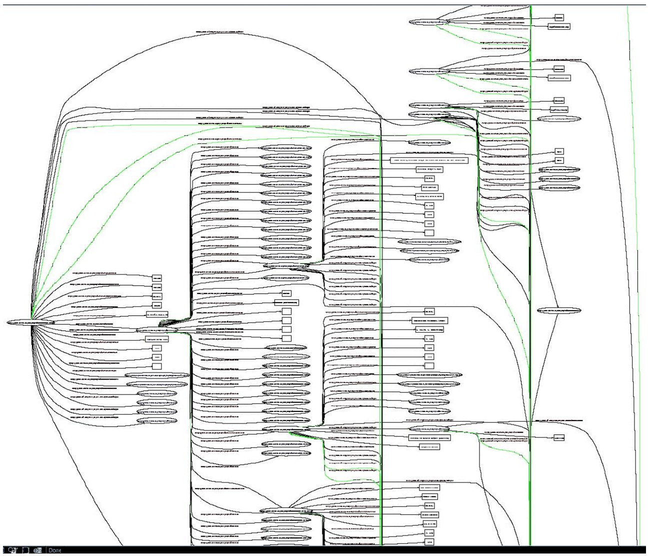


Figure 3: Small fragment of graph for ARIA RDF generated by W3C RDF Validator website

DArt$_{bio}$. Where Haystack relies on manually designed style sheets, there are also more automatic approaches. The DArt$_{bio}$ prototype, for example, generates text, graphics and layout in hypermedia presentations from an underlying database about artists [4]. Like our work, DArt$_{bio}$ demonstrates the importance and effectiveness of deriving document structure from underlying presentation-independent relational data. However, while our focus is on generating a sequential hierarchical document structure, we also generate some text and spatial structure from the derived document structure.

Hera. The Hera methodology specifies how to make systems that take RDF-encoded information in and generate navigable presentations from it [19]. Hera specifies some of the key components of the system our work presents: the input of RDF, the querying for components and the generation of presentations. With this as context, we add the clustering-based generation of document structure from the query results, with subsequent influence on the presentation generated, as part of this methodology. Another important distinction is that Hera is domain-specific, requiring human intervention and encoding to make any encountered domain presentable.

DISC. While the systems discussed above generate document structure from traditionally computable relationships, often the more compelling document structure derives more "humanistic" concerns such as discourse. Our earlier research on the DISC system explores guiding the automatic creation of coherent presentations based on discourse structures, including hierarchy [10]. DISC typically builds its presentations top-down, starting with domain-specific discourse-based general structure and then determining the lower level details. Our presentation construction, on the other hand, works bottom-up, starting with selected content and then generating higher-level, broader presentation structure such as hierarchy around it. The two systems' hierarchies differ in nature because the DISC system uses human-crafted structural templates, which can thus have richer inherent discourse. Our computer generated hierarchies, on the other hand, are simpler in discourse, but their

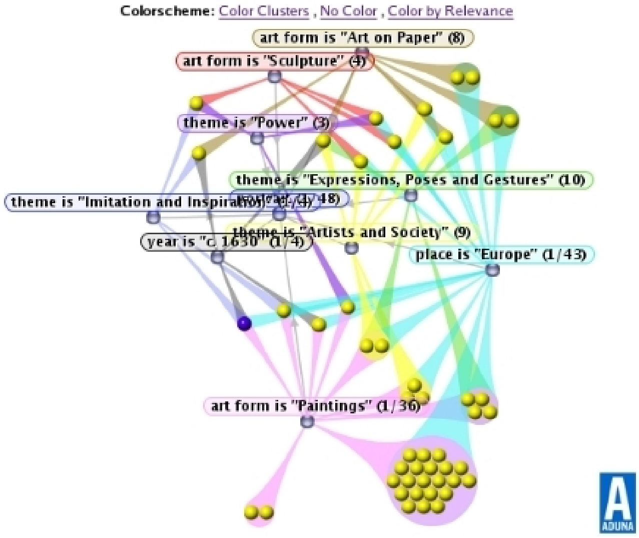

Colorscheme: Color Clusters , No Color , Color by Relevance
art form is "Art on Paper" (8)
art form is "Sculpture" (4)
theme is "Power" (3)
theme is "Expressions, Poses and Gestures" (10)
theme is "Imitation and Inspiration"
year is "c 1630" (1/4)
theme is "Artists and Society" (9)
place is "Europe" (1/43)
art form is "Paintings" (1/36)
ADUNA

Figure 4: AutoFocus generated visualization of our example RDF structure

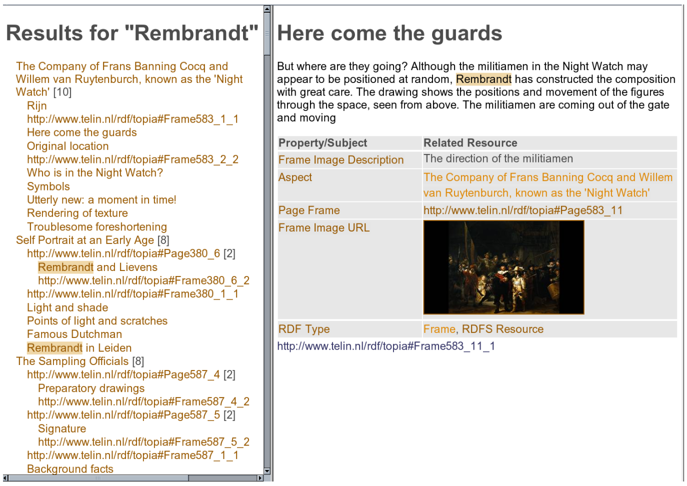
simplicity and derivation from general relation structure within semantic networks apply more readily to a wider variety of domains.

Topia. Having presented systems from other research influencing our work, we know present previous work of our own that acts as a prototype for this work: Topia. The Topia system was built as a demonstrator interface for accessing text and image resources from the Rijksmuseum ARIA database [18]. While one of its goals is to provide flexible access to the repository, the layout and interaction is typical of museum websites [8]. The Topia system enables the user to specify a query and generates the presentation automatically, including its high-level structure, from the RDF media repository. We use Topia, along with the Sesame Explore interface, as a starting-off point for the system we developed for this paper: Noadster. While Topia was written specifically for the Rijksmuseum Amsterdam, Noadster is inherently domain-independent. This domain-independence applies both to the local and global interfaces. We use Noadster as a running example throughout the rest of this paper.

Toward Noadster. This paper's main demo, *Noadster*, extends both Topia's local and global interface from ARIA to domain-independence, enabling browsing of unfamiliar repositories. Noadster illustrates potential ways of structuring information and conveying this structure, allowing the user to explore a particular view of the repository. Figure 5 shows a presentation generated by Noadster for the Rijksmuseum ARIA RDF repository. For cross-domain comparison, Figure 6 shows a Noadster presentation from the RDF repository describing our research group. We use Noadster throughout the rest of this paper as a running example of how this paper's ideas work in practice.

3. GLOBAL INTERFACE

This section discusses extending the current web search experience into the Semantic Web. A user's web experience typically begins with a search. Here, the user types in a search phrase. Then the system responds with a list of matches. The search phrase is the user's specification of a topic he or she wants a document about. The list returned is the user's global interface to the web from the perspective of that topic. This section thus discusses how the Semantic Web can apply to this search and the list of returns it generates.



Results for "Rembrandt"

The Company of Frans Banning Cocq and Willem van Ruytenburch, known as the 'Night Watch' [10]
- Rijn
- http://www.telin.nl/rdf/topia#Frame583_1_1
- Here come the guards
- Original location
- http://www.telin.nl/rdf/topia#Frame583_2_2
- Who is in the Night Watch?
- Symbols
- Utterly new: a moment in time!
- Rendering of texture
- Troublesome foreshortening

Self Portrait at an Early Age [8]
- http://www.telin.nl/rdf/topia#Page380_6 [2]
  - Rembrandt and Lievens
  - http://www.telin.nl/rdf/topia#Frame380_6_2
- http://www.telin.nl/rdf/topia#Frame380_1_1
- Light and shade
- Points of light and scratches
- Famous Dutchman
- Rembrandt in Leiden

The Sampling Officials [8]
- http://www.telin.nl/rdf/topia#Page587_4 [2]
  - Preparatory drawings
  - http://www.telin.nl/rdf/topia#Frame587_4_2
- http://www.telin.nl/rdf/topia#Page587_5 [2]
  - Signature
  - http://www.telin.nl/rdf/topia#Frame587_5_2
- http://www.telin.nl/rdf/topia#Frame587_1_1
- Background facts

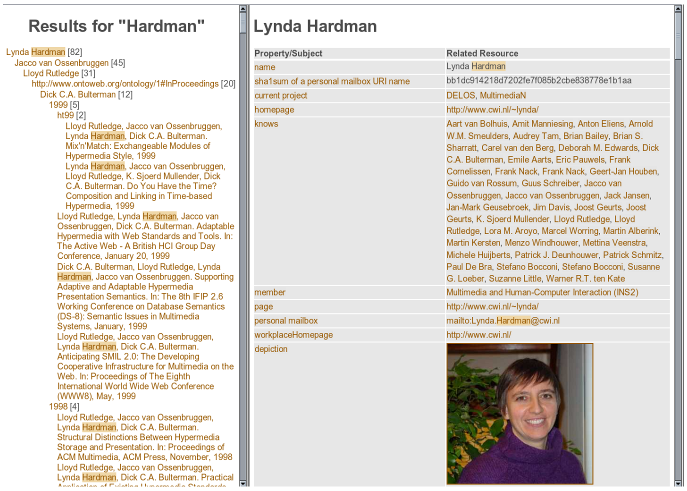
Here come the guards

But where are they going? Although the militiamen in the Night Watch may appear to be positioned at random, Rembrandt has constructed the composition with great care. The drawing shows the positions and movement of the figures through the space, seen from above. The militiamen are coming out of the gate and moving

Property/Subject	Related Resource
Frame Image Description	The direction of the militiamen
Aspect	The Company of Frans Banning Cocq and Willem van Ruytenburch, known as the 'Night Watch'
Page Frame	http://www.telin.nl/rdf/topia#Page583_11
Frame Image URL	
RDF Type	Frame, RDFS Resource

http://www.telin.nl/rdf/topia#Frame583_11_1

Figure 5: Noadster-generated presentation (image ©Rijksmuseum Amsterdam, used with permission)



Results for "Hardman"

- Lynda Hardman [82]
  - Jacco van Ossenbruggen [45]
    - Lloyd Rutledge [31]
      - http://www.ontoweb.org/ontology/1#InProceedings [20]
        - Dick C.A. Bulterman [12]
          - 1999 [5]
            - ht99 [2]
              - Lloyd Rutledge, Jacco van Ossenbruggen, Lynda Hardman, Dick C.A. Bulterman. Mix'n'Match: Exchangeable Modules of Hypermedia Style, 1999
              - Lynda Hardman, Jacco van Ossenbruggen, Lloyd Rutledge, K. Sjoerd Mullender, Dick C.A. Bulterman. Do You Have the Time? Composition and Linking in Time-based Hypermedia, 1999
            - Lloyd Rutledge, Lynda Hardman, Jacco van Ossenbruggen, Dick C.A. Bulterman. Adaptable Hypermedia with Web Standards and Tools. In: The Active Web - A British HCI Group Day Conference, January 20, 1999
            - Dick C.A. Bulterman, Lloyd Rutledge, Lynda Hardman, Jacco van Ossenbruggen. Supporting Adaptive and Adaptable Hypermedia Presentation Semantics. In: The 8th IFIP 2.6 Working Conference on Database Semantics (DS-8): Semantic Issues in Multimedia Systems, January, 1999
            - Lloyd Rutledge, Jacco van Ossenbruggen, Lynda Hardman, Dick C.A. Bulterman. Anticipating SMIL 2.0: The Developing Cooperative Infrastructure for Multimedia on the Web. In: Proceedings of The Eighth International World Wide Web Conference (WWW8), May, 1999
          - 1998 [4]
            - Lloyd Rutledge, Jacco van Ossenbruggen, Lynda Hardman, Dick C.A. Bulterman. Structural Distinctions Between Hypermedia Storage and Presentation. In: Proceedings of ACM Multimedia, ACM Press, November, 1998
            - Lloyd Rutledge, Jacco van Ossenbruggen, Lynda Hardman, Dick C.A. Bulterman. Practical

Lynda Hardman

Property/Subject	Related Resource
name	Lynda Hardman
sha1sum of a personal mailbox URI name	bb1dc914218d7202fe7f085b2cbe838778e1b1aa
current project	DELOS, MultimediaN
homepage	http://www.cwi.nl/~lynda/
knows	Aart van Bolhuis, Amit Manniesing, Anton Eliens, Arnold W.M. Smeulders, Audrey Tam, Brian Bailey, Brian S. Sharratt, Carel van den Berg, Deborah M. Edwards, Dick C.A. Bulterman, Emile Aarts, Eric Pauwels, Frank Cornelissen, Frank Nack, Frank Nack, Geert-Jan Houben, Guido van Rossum, Guus Schreiber, Jacco van Ossenbruggen, Jacco van Ossenbruggen, Jack Jansen, Jan-Mark Geusebroek, Jim Davis, Joost Geurts, Joost Geurts, K. Sjoerd Mullender, Lloyd Rutledge, Lloyd Rutledge, Lora M. Aroyo, Marcel Worring, Martin Alberink, Martin Kersten, Menzo Windhouwer, Mettina Veenstra, Michele Huijberts, Patrick J. Deunhouwer, Patrick Schmitz, Paul De Bra, Stefano Bocconi, Stefano Bocconi, Susanne G. Loeber, Suzanne Little, Warner R.T. ten Kate
member	Multimedia and Human-Computer Interaction (INS2)
page	http://www.cwi.nl/~lynda/
personal mailbox	mailto:Lynda.Hardman@cwi.nl
workplaceHomepage	http://www.cwi.nl/
depiction	

Figure 6: Same Noadster interface, this time applied to a FOAF RDF repository

3.1 Selection

Here we discuss the issue of selection from the perspective of the global interface. Specifically, this means answering the question "How does the user specify which components of the whole information store are relevant to the current information need?". First, we discuss what this means now for the WWW. Then we compare this to how global selection applies to RDF. Finally, we discuss how we implement global selection from RDF within Noadster.

WWW. Currently, the most commonly used method of finding information from the Web is to use a keyword-based query applied to the content of hypertext documents on the World Wide Web. For most users, this means typing in one or more words and getting a mostly flat list of web documents that contain them. For discussion here, we consider this set as the selection for presentation to the user.

RDF. RDF's relational knowledge structure offers additional possibilities for querying. For example, Sesame offers SeRQL

(Sesame RDF Query Language) for requesting information from RDF repositories [7]. However, proper use of this structure requires domain-specific familiarity. Fortunately, text-based search still applies to RDF because of its *literals*, which are property values consisting of straight text instead of URI references. We exploit this text-based search through RDF literals in this paper.

Noadster. Noadster provides such text-based search using literal values. The search result is represented by the indented list exemplified on the left-hand side of Figure 5, which shows the results of the query "Rembrandt". Specifically, the search is realized by a SeRQL query that searches for literal values that include the query text as a sub-string and returns all resources that appear in triples as subjects with the literal as the object. The RDF repository for ARIA includes all the original text in the database as RDF literals, enabling this type of search. While any query returning RDF resources would work, literal sub-string queries are domain-independent, not requiring advance knowledge of the RDF to encode and process. Thus, our system can also readily use more domain-specific queries. When displaying a topic node, it helps the user to show why the node matches the request. Noadster does this by highlighting the matching string in the display of the relevant literal.

3.2 Structure

This paper's most significant contribution to search returns is the generation of document structure. Human authors structure information by grouping related information together, typically in a recursive fashion resulting in a hierarchy. This hierarchical structure traditionally appears as sections and subsections. Such structuring helps readers see relationships between different pieces of information that would otherwise remain unnoticed. Presenting hierarchical groups often involves adding introductory sections around a group's subsections. More noticeably, traditional document structure also enables Tables of Contents as facilitated access to document content. This section aims to generate section introductions and Tables of Contents for Semantic Web search returns.

While the selection techniques described in the previous subsection greatly reduce the number of immediately accessible resources to the most relevant ones, this number can still be cumbersomely large, as it frequently is for web document searches. We address this by grouping these matching resources into a hierarchy as the basis of document structure. By asserting that systems can generate document structure as a topic-focused interface to an RDF repository, this subsection is the core of this paper.

RDF repository structure is, for the most part, an unstructured node-edge graph. Document structure, on the other hand, is a sequenced hierarchy. Transformation between the two must account for this difference. Here, we discuss how this transformation occurs in terms of the core components of document structure we identify in early work [18]: hierarchy, sequence and cross-reference.

Hierarchy

Hierarchy is the cornerstone of document structure. It communicates that certain objects relate to each other by grouping them together. Importantly, good hierarchical documents also clarify the nature of the grouping, discussing the nature of the group itself as well as that of its components.

Concept Lattice Clustering. Noadster uses concept lattice clustering to build its hierarchies over search returns. Our algorithm here is adapted from the concept lattice clustering Topia uses [18]. In our experience with using Noadster, this generates rich and informative hierarchies. The concept lattice algorithm identifies *characteristics* of the components to cluster and puts components with identical information into the same groups. This grouping is nested and thus hierarchical. Specifically, in RDF terminology, the characteristics that Noadster clusters on are predicate-object pairs in triples with the characterized component as the subject. As such, any resources directly linked, in either direction, to a common resource are subject to clustering.

Enhancing Clustering with Subclasses. The more characteristics resources have, the more possibilities there are for grouping them. The `rdfs:subClass` predicate lets the writer of RDF code easily enhance this clustering by having RDF processing infer extra characteristics beyond those directly encoded. Making one class a subclass of another causes any predicate-object assignment to the first class to be effectively copied to an additional predicate-object assignment the second. This is recursive, making a predicate to a class also be a predicate to all of its super-classes. These extra predicate-object assignments provide extra characteristics to cluster upon. They also enable clustering to generate more levels in the resulting hierarchies. For example, we use subclasses in the ARIA RDF by encoding a hierarchy of genres as genre concept resources with `rdfs:subClass` predicate to their parent genres, making genres a strong component of the generated multi-level hierarchies.

Introduction Screen Displays. It is not enough to show what falls in a group. A system should also show why. One thing we found lacking in Topia was explanation of the groups. Topia presents each group in the hierarchy only an outline display item. Screen displays only appear for the matching resources. To address this, Noadster presents screen displays for groups as well. We call these *introductions* because they function similarly to the introductions typically found for sections in books and articles. They describe a group as a whole before going into the details of its components, thus helping communicate to the user what significance the groups have and what insights they give about the relations between their contents. Specifically, for each group, Noadster generates a screen display whose focal point is the resource URI for the object in the predicate-object assignment making up the group's common characteristic.

Introduction Sections. Sometimes a group shares more than one characteristic. In Noadster, this results in multiple screen displays for an introduction. The system handles this by making a group for the introduction that appears as an additional subsection. This resembles introductions subsections in text books, as compared to introductions consisting of a few paragraphs with a header.

Sequence

Like hierarchy, sequence is a core component of document structure. The sequence in which components of a document appear often communicates important insights about the relation between them. Web search engines sequence their returns based on relevance measures, placing the most likely matches toward the front of the list. Here, the sequence if more functional than informative. Noadster performs similar sequencing by sorting subgroups of a common parent based on how many matching resources they contain, making the groups with the most content relevant to the topic request appear earlier.

Topia, on the other hand, provides informative sequencing by sorting artifacts within the same group by year of creation [18]. This sorting, however, is quite domain-specific. The domain-independent components of Noadster do not have the benefit of such knowledge about which properties in a given repository generate meaningful sequence.

Structural Style

The means of structuring presented above constitutes one "style" for transforming RDF to document structure. However, there is a wide variety of other potential "structural style sheets" for doing so. Here, we discuss these possibilities. We also discuss means for authors, designers and users to set such style parameters for storage and processing.

Predicate Weights. In a given domain, and for a given user, some concepts are more important than others for generating document structure. Thus it is helpful to specify for a given domain, and possibly also for a given user using that domain, how important each concept is. Topia lets users specify style for generating document structure from RDF [18]. Here, users specify weights of significance for a selection of RDF predicates from the ARIA RDF. This allows the concept lattice algorithm to recognize smaller clusters as significant enough to form hierarchical groups if the predicates that form them are more significant than competing larger groups sharing less important properties. Similarly, smaller sized groups can appear sequentially before larger groups. Since Topia gives all predicates a default middle score, users do not need to specify style to start surfing a new RDF repository. As they grow familiar with a repository, they can incrementally adjust the scores of one or more predicates.

Domain-independent Predicate Weights. While Topia's list of such properties is hard coded and thus domain-specific, a system could easily generate a list of all RDF resources used as predicates and place it in the same type of interface. The primary difficulty comes when a repository has very many predicates. However, because all predicates start with a default middle score, this domain-independent adaptation will still generate document structure initially and then allow the user to incrementally improve the clustering style. We plan to apply this technique to Noadster.

Beyond Concept Lattices. While Noadster uses one means of clustering, that of concept lattices, an RDF style sheet designer has many means available for designing the derivation of document structure. In earlier work, we present several categories of clustering techniques for generating document structure that apply to our system, including property, relation and axial clustering [2]. Despite this wide variety of techniques, the core components of the output structure remain the same. Essentially, all these techniques can output the XML format we made for document structure, which then feeds into to Noadster global interface to generate a presentation.

Further User Control. We hope to extend the user-as-author paradigm by providing users additional control over the "style" of presentation generation. This includes not just control over more aspects of the generation but also quickened feedback-like control for incrementally modifying generation paradigms during presentation time. Such control invites integration with the SampLe system [9], especially user-tuned selection and the selection of genres as structural styles.

3.3 Surfability

Because search returns, be they in flat lists or generated hierarchical structure, have links to the corresponding matches, users can navigate quickly to the documents to see which serve their needs. Furthermore, when search returns fall in a hierarchical structure, the user can navigation the hierarchy itself to find entire groups of matches that the user's sees as particularly appropriate. This large-scale navigation of the document structure also conveys to the user the additional insight about conceptual relationships between the matching components, making the global interface itself a unified document about the given topic.

Noadster identifies search result groups as names. These name displays are also links to corresponding repository components. As such, clicking such a group name displays the group as the current focal point. The focal point resource is highlighted in the global view outline.

4. LOCAL INTERFACE

The previous section discussed how the Semantic Web can change the search for information and the structuring of returns. On the text-based World Wide Web, such structured lists of search returns provide direct navigation to documents, which typically make up the URI-defined information unit for web search engines. This section discusses the displays this paper's Semantic Web global interface provides access to. As with the WWW, these are displays for particular URI's. However, as we discuss here, what makes up a display for a URI is much different for the Semantic Web than for the traditional web.

The localized level of the user interface presents information regarding a single component in the repository. This is a local, small-scale perspective of the user's current place in the navigation provided. This section describes this concept of location, the accessing of media associated within and the structuring of this media's display.

The Noadster demonstrator makes no use of domain-dependent knowledge, and does not need any *a priori* knowledge of the RDF vocabulary used. For example, Figure 6 shows the same Noadster interface, applied to another RDF repository, this time containing FOAF[1] data describing the social network of the people in our research group.

While showing the subset of the repository the user is interested in, the presentation should also show the relationship with the rest of the repository, either at the local, focal point, level or the global, repository-wide, level. We advocate that different scales of interface can be merged with each other in a way that enhances the user's understanding of both the overall content of the RDF repository and their understanding of the local neighborhood of the current focal point. The structures we aim to convey are the relationship of the focal point to the user's specified area of interest, a user-centric overriding structure to retain manageability and how the area of interest relates to the rest of the repository.

4.1 Selection

Selection via search puts the end-user in a position similar to the author: where the author needs search to find the raw materials for the document that needs to be written, the end-user need to find those documents that are needed to fulfill the current information need. For many current search tools, virtually all returning a ranked list of document pointers, the difference between the two is not apparent. Some more advanced tools, however, try to move from returning links to relevant documents to returning a single, standalone, document that directly provides the answer to the [6].

A key concept in our model is that of the *focal point* in the network. The focal point can be any node or edge in the RDF graph, specified by a URI. Here, a focal point is not so much a body of information but the hub of potentially many triples spanning out from it that collectively provide information. The focal point itself is devoid of content. The media conveying it comes from its triples. The system thus selects media content from these triples.

Associated triples. Since the repository contains triples as the basic unit, there must be at least one triple associated with the current focal point. We identify the *associated triples* as the set of

[1] `http://www.foaf-project.org/`

triples including the focal point as either object or subject. These convey information about the focal point. Specifically, they convey its properties and other WWW resources or other RDF concepts that have a direct relationship with the focal point. While viewing the current focal point, the user has direct access to all resources sharing a triple with the focal point.

Literals. While the RDF repository is mostly a collection of triples using URIs as nodes and edges of the graph, some triples involve literal objects. Since these consist of plain text, they are much better suited for direct display in the presentation than URIs. As shown in Figure 5, the direct display of literal content is typically not problematic, as most RDF literals are relatively small pieces of text.

Labels. A particularly informative type of literal is the `rdfs:label`. RDFS defines this as "a human-readable version of a resource's name" [20]. The Sesame Explorer interface has an option that replaces all URIs in the interface with their associated labels, when available. Noadster does this as well, making the text in each link to another resource contain that resource's `rdfs:label`. Noadster also goes further by making such labels titles for screen displays of resources, displaying them at the top in large, bold font, as shown in Figure 5. Thus Noadster treats `rdfs:label` as the initial means of conveying a resource to the user, be it the current resource or an immediately traversable one.

Label Sub-properties. Sometimes an RDF repository may have domain-specific predicates that can function as labels. Authors of such can encode this as making each of these predicates a `rdfs:subProperty` of `rdfs:label`. They can encode this once in the RDFS for each predicate type, and each instance will resultantly have that predicate as its label. In the ARIA repository, for example, the names of artists are made sub-properties of `rdfs:label`. With this single RDFS assignment, all artist names become recognized as labels, and thus displayed as such by Noadster.

External Media. In RDF, properties values that are not literals are URIs. URIs often reference directly presentable media items, which, like literals, systems can integrate into screen displays. Noadster performs such direct integration of external image media. When a focal point shares a triple with such an image resource, Noadster presents it directly in the focal point's display. This allows the user to see that there is an image and simultaneously what that image is. Noadster applies a number of strategies to find out the MIME type of a resource, and tries to use this information to improve the display. In Figure 5, for example, the image resources related to the painting appear directly in the associated triples.

4.2 Structure

Allowing access to all the URIs related directly to the current focal point provides a user interface challenge if there are many triples associated with the focal point. The layout should group the triples to provide an overview of the different types of information related to the focal point, and to allow the user to find quickly the type of information that satisfies the current information need. Here, we present first the simple, default means with which Noadster groups a focal point's triples into its spatial layout. Then we discuss briefly how to extend this with structuring techniques presented in Section 3.

RDF(S)-based Grouping. One method Noadster uses is to group triples by shared subject, predicate or object value. The grouping methods so far are based on distinctions that can be seen in the "flat" RDF graph structure defined by directly encoded triples. More information can be gleaned by also looking at additional triples inferred by the predefined semantics of the `rdfs:subClass` and `rdfs:subProperty` hierarchies. These can provide explicit grouping information of the repository that allows hierarchical structures to be identified.

Clustering Triples. The global clustering techniques we discussed in Section 3 can apply to structuring local display spatial layout as well. This clustering groups items based on shared characteristics. Specifically, this shared characteristic is the RDF predicate-object in triples for which multiple resources each act as one of the triple's subject. In Noadster's case, we already mentioned the namespace of the triple's predicate as an important clustering characteristic, as well as common single triple roles. Other potential characteristics include the namespace of the subjects and objects and the role the focal point plays in the triples. Such clustering lets Noadster determine the grouping strategy that puts most related triples together. It also allows more levels of depth in the grouping, providing a document hierarchy. Finally, user selection of clustering strategies also applies to spatial Noadster clustering as well.

Multiple Displays for Single Focal Points. While grouped layout helps users sort larger amounts of information, resources in some RDF repositories can easily involve far more triples than can appear in a single display. A potential solution is applying clustering techniques to group triples into separate displays rather than just separate screen areas when they become too large. From the user's perspective, these separate display units would seem the same as resource focal points.

4.3 Surfability

Given the generation of well-organized displays as described above, the next consideration is where to go next. RDF browsers, including Sesame explore mode and Noadster, usually place links in the current display that lead to other displays. Both these systems display two components for each triple involving the focal point. Each component that is not a literal references another RDF resource. Whatever Noadster displays for each one, whether it is its label, image or other media, the user can click on it to generate the display with that component as the focal point.

Lost in Semantic Space. Given these two links for each triple, the user may have an overwhelming number of choices for each local display. Furthermore, the local interface enables eventual navigation through the entire RDF repository, offering an overwhelming number of potential current locations. Given all this, users can become easily lost. Therefore, users should always be able to retain some notion of where they are in relation to the rest of the repository [13]. While surfability is important for giving the user full control, some guidance is essential for helping the user make appropriate choices and to understand where he or she sits in the repository as a whole.

Current Location. Coordination between the local and global interfaces provides such guidance. One important coordination is an indication of which entry in the global interface is the current focus — that is, of course, when the current location is in the match list. Noadster conveys this by highlighting, by default in yellow, the entry in the global interface that corresponds with the current local display.

Showing Cross-References. In addition to showing the current location in the global view, it also helps to show the current location's neighboring components that are in the global view. Noadster communicates by highlighting, by default in orange, those entries in the global view that are direct neighbors of the current location. Links to these direct neighbors in the hierarchy also have highlights in Noadster's local interface.

Generating Cross-References. Concept lattice clustering can place some groups in multiple places in the generated hierarchy, making the generated document structure more of a directed acyclic graph. Earlier work on Topia calls these *recurrences* [18], presenting them as single matches that appear in multiple locations. Noadster presents these instead as cross-references, in the manner described above.

5. CLOSING

This paper describes making meaningful presentations from RDF-annotated media repositories. This opens up the Semantic Web as a whole to immediate access from any user using one system. It also serves as a foundation for "semantic style" that content providers can make for specific domains and users can make for specific domains as well as for RDF-derived displays in general. Here we wrap up this paper by presenting its overview, insights and concluding remarks.

Overview. The type of system we present allows content providers to define networks of related concepts and media items from which end users can request tailor-made hypermedia presentations. Such systems can have a readily extensible domain-independent foundation, providing immediate generalized access to unfamiliar domains for users and quick improvement to this access from document structure engineers. We discuss allowing end users to specify topics for guiding navigation through RDF-annotated media repositories. Localized display generation for individual components in the RDF encoding provides basic access and navigation. The generated interface emphasizes and facilitates access to information relevant to the topic requested. As part of this, clustering algorithms on these selected components generates document structure around them, giving them informative context in the generated presentation as a whole. The result provides tailored hypermedia presentation generation on request for a given RDF-annotated media repository.

Future Work: Domain-specific Extensions. Having established a generalized foundation for domain-independent RDF-annotated media access, the logical next step is exploring its extension into different document genres, keeping the domain-independent functions as a foundation that provides commonalities to all domain interfaces while facilitating development of the domain-specific aspects of each. With this in mind, Noadster treats its domain-independent component not as the whole package but instead as a starting point for adding domain-specific extensions. The XSLT code defining Noadster allows inclusion of external XSLT files defining presentation for specific domains, starting with Dublin Core. This plug-in approach adds these domain-specific sub-displays to the main display for each node. This generates a vertical sequence of displays for each domain included. In addition to these domain-specific extensions to the focal point display, potential exploration includes developing new structure building strategies derived from more developed discourse techniques such as those in DISC [10], resulting in richer presentations from the human perspective.

Insights. This perspective of search engines as retrieving content rather than documents makes them on-demand generators of new presentations rather than retrievers of existing documents. The key difference between search engines and typical presentation generation is the granularity of their components. Search engines typically return entire documents, which have multiple components and internal structure and can be very large. Hypermedia presentation generation, on the other hand, typically handles individual media objects and small clips or single words of text. This finer granularity greatly liberates the possibilities for the generated documents far beyond the confines of what document structure already exists in human-written documents.

Conclusion. While much of this paper's description of its system suggests a "magic bullet" application making RDF as presentable and popular as HTML, its results will instead naturally have the clunkiness that computer generation makes. Our working assumption to overcome this is that user approaches to web search engine results can also apply here. That is, while search results are of course much poorer than what a human expert librarian would return for a document request, they have nonetheless become a main entry way for the WWW. This is because lay users have quickly learned to use what the computer provides and see around the computer glitches. Our challenge is to translate this user approach from document search to document structure, making this paper's system a general-purpose portal to the Semantic Web as a whole. While making sensible document structure is an ability typically considered to lie on the far side of the Artificial Intelligence boundary, our hope is that taking simple assumptions and a simple model and processing them in bulk will generate enough sense to be quite useful.

Further Resources. The demos and other resources for this paper are accessible at `http://homepages.cwi.nl/~media/conferences/WWW2005/`.

Acknowledgments

This work was funded in part by the Dutch national ToKeN2000/I^{2}RP, NWO/NASH and Senter/MultimediaN projects. We thank the Rijksmuseum Amsterdam for their permission to use their website's database and media content [17]. Martin Alberink and Mettina Veenstra of the Telematica Instituut contributed ideas regarding clustering and document structure. Geert-Jan Houben gave useful suggestions in motivating and structuring this work, and offered essential insight into the form of the paper itself. Patrick Schmitz provides us regular uniquely useful insights into the Semantic Web, which helped form this paper. Finally, we thank our CWI colleagues Frank Nack, Joost Geurts, Stefano Bocconi, Katya Falkovych and Katharina Schwarz for countless discussions and constructively devastating comments on this work.

6. REFERENCES

[1] Aduna B.V. Autofocus Personal. http://aduna.biz/products/autofocus/personal/index.html, 2004.

[2] M. Alberink, L. Rutledge, L. Hardman, and M. Veenstra. Clustering Semantics for Hypermedia Presentation. Technical Report INS-E0409, CWI, November 2004.

[3] L. Aroyo and R. Denaux. ClusterMap Visualization for Topia. http://wwwis.win.tue.nl:8082/clustermap/index.html, 2004.

[4] J. Bateman, J. Kleinz, T. Kamps, and K. Reichenberger. Towards Constructive Text, Diagram, and Layout Generation for Information Presentation. *Computational Linguistics*, 27(3):409–449, September 2001.

[5] T. Berners-Lee. Notation 3: An RDF Language for the Semantic Web. W3C Design Issue, 1998.

[6] M. Bordegoni, G. Faconti, M. Maybury, T. Rist, S. Ruggieri, P. Trahanias, and M. Wilson. A Standard Reference Model for Intelligent Multimedia Presentation Systems. *Computer Standards & Interfaces*, 18(6-7):477–496, December 1997.

[7] J. Broekstra, A. Kampman, and F. van Harmelen. Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema. In I. Horrocks and J. Hendler, editors, *The*

Semantic Web - ISWC 2002, number 2342 in Lecture Notes in Computer Science, pages 54–68, Berlin Heidelberg, 2002. Springer.

[8] K. Czajka. Design of interactive and adaptive interfaces to exploit large media-based knowledge spaces in the domain of museums for the fine arts. Diploma Media System Design, University of Applied Sciences Darmstadt, June 14, 2002.

[9] K. Falkovych, J. Werner, and F. Nack. Semantic-based Support for the Semi-automatic Construction of Multimedia Presentations. Position paper for the First International Workshop on Interaction Design and the Semantic Web held in conjunction with the World Wide Web Conference, 2004, NYC, May 18, 2004.

[10] J. Geurts, S. Bocconi, J. van Ossenbruggen, and L. Hardman. Towards Ontology-driven Discourse: From Semantic Graphs to Multimedia Presentations. In *Second International Semantic Web Conference (ISWC2003)*, pages 597–612, Sanibel Island, Florida, USA, October 20-23, 2003.

[11] N. Gibbins, S. Harris, and m. schraefel. Applying mSpace Interfaces to the Semantic Web. Technical Report 8639, Electronics and Computer Science — a school of the University of Southampton, November 2003.

[12] W. Grosso, H. Eriksson, R. Fergerson, J. Gennari, S. Tu, and M. Musen. Knowledge Modeling at the Millennium (The Design and Evolution of Protege-2000). Technical Report SMI Report Number: SMI-1999-0801, Stanford Medical Informatics (SMI), 1999.

[13] L. Hardman and B. S. Sharratt. User-Centred Hypertext Design: The Application of HCI Design Principles and Guidelines. In R. McAleese and C. Green, editors, *Hypertext: State of the Art*, pages 252–259, York, UK, 1989.

[14] F. Manola and E. M. (ed.). RDF Primer. W3C Recommendation, 2004.

[15] E. Prud'hommeaux and R. Lee. W3C RDF Validation Service. http://www.w3.org/RDF/Validator/.

[16] D. Quan and D. R. Karger. How to Make a Semantic Web Browser. In *The Thirteenth International World Wide Web Conference*, Ney York City, NY, May 17-22, 2004. IW3C2, ACM Press.

[17] Rijksmuseum Amsterdam. Rijksmuseum Amsterdam Website. http://www.rijksmuseum.nl.

[18] L. Rutledge, M. Alberink, R. Brussee, S. Pokraev, W. van Dieten, and M. Veenstra. Finding the Story — Broader Applicability of Semantics and Discourse for Hypermedia Generation. In *Proceedings of the 14th ACM conference on Hypertext and Hypermedia*, pages 67–76, Nottingham, UK, August 23-27, 2003. ACM.

[19] R. Vdovjak, F. Frasincar, G. Houben, and P. Barna. Engineering Semantic Web Information Systems in Hera. *Journal of Web Engineering*, 2(1 and 2):3–26, 2003.

[20] W3C. RDF Vocabulary Description Language 1.0: RDF Schema. W3C Recommendation, 10 February 2004. Edited by Dan Brickley and R.V. Guha.