Centrum voor Wiskunde en Informatica

REPORTRAPPORT

INS

Information Systems

INformation Systems

Making RDF presentable: integrated global and local Semantic Web browsing

L.W. Rutledge, J.R. van Ossenbruggen, L. Hardman

REPORT INS-E0505 MARCH 2005

CWI is the National Research Institute for Mathematics and Computer Science. It is sponsored by the Netherlands Organization for Scientific Research (NWO).
CWI is a founding member of ERCIM, the European Research Consortium for Informatics and Mathematics.

CWI's research has a theme-oriented structure and is grouped into four clusters. Listed below are the names of the clusters and in parentheses their acronyms.

Probability, Networks and Algorithms (PNA)

Software Engineering (SEN)

Modelling, Analysis and Simulation (MAS)

Information Systems (INS)

Making RDF presentable: integrated global and local Semantic Web browsing

ABSTRACT

This paper discusses generating document structure from annotated media repositories in a domain-independent manner. This approaches the vision of a universal RDF browser. We start by applying the search-and-browse paradigm established for the WWW to RDF presentation. Furthermore, 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 required document has not yet been written. The resulting system allows a user to specify a topic for which it generates a hypermedia document providing guided navigation through virtually any RDF repository. The impact for content providers is that as soon as one adds new media items and their annotations to a repository, they become immediately available for automatic integration into subsequently requested presentations.

Making RDF Presentable:

Integrated Global and Local Semantic Web Browsing

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 universal RDF browser. We start by applying the search-and-browse paradigm established for the WWW to RDF presentation. Furthermore, 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 required document has not yet been written. The resulting system allows a user to specify a topic for which it generates a hypermedia document providing guided navigation through virtually any RDF repository. The impact for content providers is that as soon as one adds new media items and their annotations 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

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 with general-purpose browsers. 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 RDF lacks the document structure HTML and other XML formats have: primarily, that of hierarchy and sequence. Hierarchy and sequence have long been cornerstones of document structure. Human authors make large amounts of information more readily accessed and learned by readers by grouping it and sorting it in meaningful and insightful ways. A core aspect of XML is that it lets writers focus on the hierarchy and sequence of their documents as independent of any subsequently rendered presentation.

Of course, RDF intentionally lacks hierarchy and sequence, choosing instead to facilitate machine-processing of the assertions it encodes. However, this focus on machine-processing 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 users. Converting RDF structure to 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.

Our goal is to generate navigable structures that orient the user in the current local context, communicates the overall structure from this perspective and provides navigation through it while maintaining a sense of orientation 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. We can thus help improve the readability of lists of RDF statements by ordering, grouping and prioritizing them before presentation.

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 InterActief) database, which drives the interface to its collection website [16]. ARIA includes about 1250 artifacts from the mu-

---

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

seum, associating them not just with images but also with concepts such as description, genres, detail and artists.

After a review of related work in Section 2, Section 3 explores the determination of an RDF-derived presentation's overall document structure. In Section 4 we describe the generation from RDF of the individual screen displays that make up a presentation. After that, Section 5 pulls these last two sections together by discussing the unification of the interfaces they present. 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 relationships among these chunks, a user might succeed without the help of an interface that makes the underlying structure explicit, such as a site-map or fish-eye view. With RDF, the situation is typically the exact opposite: we have many small chunks of information with many explicit relationships among them. The user interfaces of many RDF tools clearly reflect aspects of this observation. Here we look at interfaces that give a global view of these many relationships, then those that concentrate on a single piece of information in the space. We then review existing systems that combine these.

2.1 Global Interface

Several systems provide large-scale views of RDF repositories. These large-scale viewers focus on the broad relational structuring joining the content. Precisely because the emphasis is on the global structure, systems typically have poor presentation of the detailed content.

RDF Graph Generation. The most generic, both in terms of visual technique and the domains it applies to, global interface to RDF is probably the W3C's RDF Validator [14]. This system provides a graph-based interface to any RDF repository. Figure 1 shows the hyperlinked SVG version of such a graph. It automatically generates a graph-based view of the validated statements. While this gives the 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 statements, the graph quickly becomes unmanageably large.

AutoFocus. An example of a more interactive alternative for navigating structure appears in Figure 2 [3]. This diagram results from running the ARIA RDF store through an adapted version of AutoFocus [1, 8]. 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 interface renders resources as yellow dots and, except for a few labels, shows no textual content. In contrast, the W3C RDF Validator shows every URI and every literal that constitute the RDF statements 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 statements.

mSpace. mSpace [11] derives global structure for exploring relational data stores, including those encoded in RDF. Unlike our approach, it uses a multi-dimensional grid rather than 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
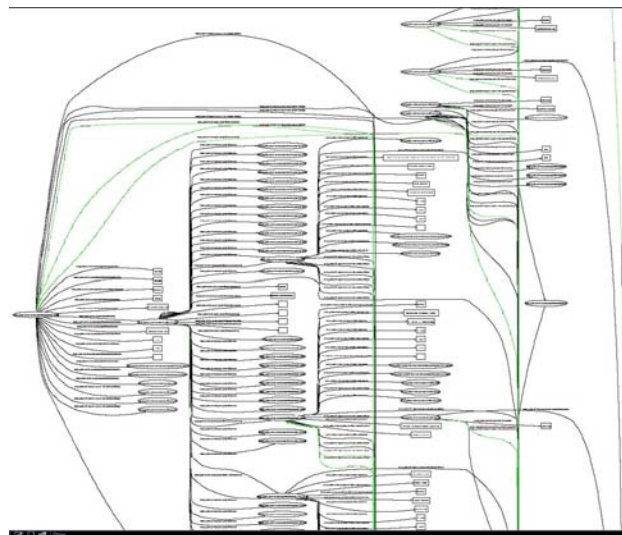

Figure 1: Small fragment of graph for ARIA RDF generated by W3C RDF Validator website

cells in each column from left to right, the user specifies incremental subsets that have those cells' property assignments. Users can change this column order. mSpace's focus is at a higher level in the information structure than addressed in this paper, 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. mSpace is also domain-specific, although it provides means of extension into any RDF repository.

2.2 Local Interface

In contrast to global interfaces which emphasize emerging structures within the relationships, a local view provides richer details for a particular information item. Users typically need information that is at this level of specificity. Local interfaces can have hyperlinks to each other, providing users with navigation through entire repositories, albeit with potentially very many traversal steps.

Sesame's 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 3 [5]. Given a particular URI, Sesame's explore mode shows all RDF statements with that URI as a subject, property or value. A link from each component generates an equivalent page for that URI, thus making RDF browsable. The current view is always limited to the immediate vicinity of the current resource. Additionally, by producing a flat list of statements, Sesame's explore mode does not show any underlying structure.

Protege-2000. Semantic Web editors such as Protege-2000 [12] offer hierarchical browsing facilities. 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 Protege-2000 also provides is similar to Sesame's explore mode navigation among statements.

2.3 Integrated Interface

While a large-scale view is comparable to exploring a forest from an airplane, with no way to land, small-scale browsing is like missing the forest for the trees. These two approaches are combined in most traditionally formed documents. An overview of the structure
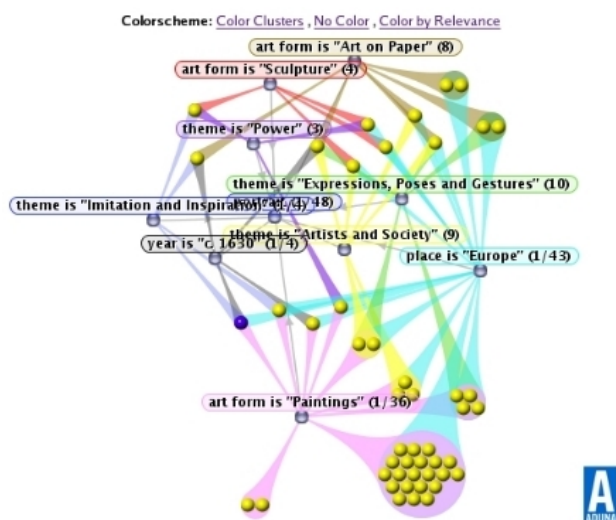
Figure 2: AutoFocus generated visualization of our example RDF structure

Figure 3: Display from Sesame's explore mode interface

is optionally given at the beginning (the table of contents) and then the different levels of structure are signaled within 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 [15] is, at the time of writing, the most well known approach to viewing RDF as a document. 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.

Simile. The Simile project [1] provides some RDF-based user interface tools, including Hayloft, a more lightweight follow-up of Haystack, and a suite of web-based RDF browsers called Longwell. The Longwell suite has many things in common with the type of browser this paper proposes. Both run server side as a Java web application and both shared Simile's stated purpose " to be able to browse and search arbitrary RDF datasets, also to prototype different user interface scenarios that could be useful to end-users, to digital librarians and to metadata analysts."[2] However, both the global and local displays of Longwell browsers are tuned to specific domains and their related schemas. This domain-specificity also applies to the generation of the structure that the global interface shows.

DArt$_{bio}$. Where Haystack and Simile rely 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]. Similar to 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 transform RDF-encoded information into navigable presentations [18]. 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 presented above generate document structure from traditionally computable relationships, often the more compelling document structure derives from more "humanistic" considerations such as discourse. Research on DISC 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. This paper's 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 simplicity and derivation from general relational structure within semantic networks apply more readily to a wider variety of domains.

Topia. Previous work of our own that acts as a prototype for this work is the Topia system. This was built as a demonstrator interface for accessing text and image resources from the Rijksmuseum ARIA database [17]. While one of its goals is to provide flexible access to the repository, the layout and interaction is typical of museum websites [6]. Topia enables the user to specify a query and generates the presentation automatically, including its high-level structure, from the RDF media repository.
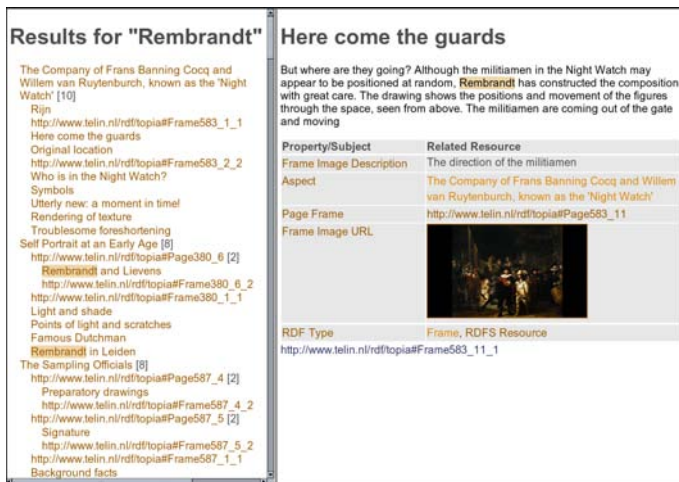
Figure 4: Noadster-generated web interface to ARIA (image ©Rijksmuseum Amsterdam, used with permission)

Figure 5: Noadster interface to CWI's MM&HCI people and publications repository

Toward Noadster. We use Topia and Sesame's explore mode as starting-off points for the system we developed for this paper: *Noadster*. While Topia was written specifically for the Rijksmuseum Amsterdam, Noadster is inherently domain-independent for both to the local and global interfaces, enabling browsing of unfamiliar repositories. Noadster illustrates potential ways of structuring information and conveying this structure, allowing users to explore different views of their repositories. Figure 4 shows a presentation generated by Noadster for the Rijksmuseum ARIA RDF repository. For cross-domain comparison, Figure 5 shows a Noadster presentation from the RDF repository describing our research group. In both figures, the global interface is on the left and the local interface is on the right. 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 typing in a search phrase. The system then responds with a list of matches. This list is the user's global interface to the web, or other repository, from the perspective of the search. For the user, this pattern of interaction remains basically the same with our approach for the Semantic Web as it has with the WWW. The underlying processing is, of course, quite different.

3.1 Selection

The most important contribution of a web query is that it specifies a subset of web pages from the much larger set of information sources available. We consider this subset as the *selection* for presentation to the user. Here we apply and adapt the familiar World Wide Web text-based selection process and its domain-independence to the Semantic Web.

Domain-independent WWW Text Search. Text-based search on the World Wide Web applies to any posted web document, regardless of who wrote it or who its audience is. Documents only need to be accessible on the WWW and in particular formats, which typically include HTML. This all-encompassing aspect of web search is perhaps taken for granted due to its success.

Domain-dependent RDF Structure Search. RDF's relational knowledge structure offers additional possibilities for querying.
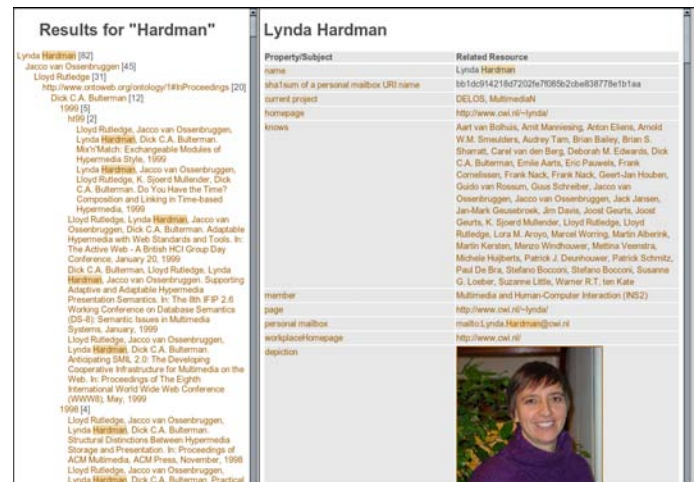
For example, Sesame offers SeRQL (Sesame RDF Query Language) for requesting information from RDF repositories [5]. However, proper use of this structure requires domain-specific familiarity. While, as we describe later, this paper relies on RDF-defined structure for domain-independent generation of both local displays and global structure, we can offer no domain-independent manner to use knowledge structure for search.

Domain-dependent RDF Literal Search. Fortunately, text-based search still applies to RDF because of its *literals*, which are property values consisting of strings instead of URI's. RDF query languages such as SeRQL provide queries that examine literal text content. Text-based search thus enables domain-independent querying over RDF repositories. Noadster provides such text-based search. The left side of Figure 4 shows the search results of the query "Rembrandt" applied to the ARIA repository. Specifically, the search is realized by a SeRQL query for literals that include the query text as a sub-string and returns all resources used in statements containing that literal.

Other Search Forms. Other types of search for RDF-encoded resources exist as well. WWW text search, for example, can apply to the text-based documents that an RDF repository annotates. Such search checks the contents of these documents for matches and returns their URIs, as with current search typified by Google. Similarly, and more broadly, feature-based search of any media would return URIs of matching media resources. Furthermore, the user can also perform selection by hand, possibly in conjunction with automated search. Regardless of what the selection process examines, be it RDF structure, RDF literals, document text, media features or user interaction, the result is a set of URIs. The structuring strategies we describe next apply to any set of RDF-annotated URIs.

3.2 Generating Structure

While search-based selection is important for accessing large repositories, this paper's focus is on building a helpful and informative structure around the returns. 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. 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. In order to transform RDF's typed link structure joining the search matches into a meaningful hierarchy for an end-user, we need to identify which explicit semantic structures can be used as a basis for human-interpretable information. Here we explore some domain-independent techniques, which allow more direct application to multiple repositories.

Structural Transform. RDF structure is a node-edge graph. Document structure, on the other hand, is a sequenced hierarchy. Transformation between the two must account for this difference.

Concept Lattice Clustering. The concept lattice clustering algorithm is one potential means of transforming semantic structure to document structure. This technique identifies *characteristics* of selected components and puts components that share characteristics into the same groups [9]. This grouping is nested and thus hierarchical. RDF descriptions provide potential cluster characteristics. Topia applies concept lattice clustering to RDF annotations to build document structure by treating RDF property value assignments as characteristics of their subjects [17]. Noadster extends this with a broader definition of a characteristic of a component: anything linked to it in either direction by any statement. This broader definition provides more characteristics, which brings more possibilities for clustering and building document structure.

Inferencing. The more characteristics resources have, the more ways there are for grouping them. The Semantic Web provides several ways of *inferring* additional characteristics from those explicitly encoded. One of these is the `rdfs:subClass` property, which causes all property-value assignments of one component to be effectively copied to another. These extra property-value assignments provide extra characteristics to cluster upon. Subclasses are recursive because a property of a class is inherited by all its descendant classes. This recursion enables clustering to generate more levels in the resulting hierarchies. We use subclasses in the ARIA RDF by encoding a hierarchy of genres as genre concept resources with `rdfs:subClass` property of their parent genres, making genres a strong component of the generated multi-level hierarchies.

Relevance Sequencing. Just as hierarchy, sequence is a core component of document structure. The sequence in which components of a document appear often communicates important insights about the relationships between them. Web search engines sequence their returns based on relevance measures, placing the most likely matches towards the front of the list. Here, the sequence is more functional than informative. Noadster performs 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.

Semantic Sequencing. While sorting by relevance can be useful, clearly the sequence of components in documents is typically based on something more meaningful. Document sequence, like hierarchy, communicates relationships between components. Topia derives meaningful sequence from the underlying meta-data by sorting artifacts within the same group by year of creation [17]. 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.

3.3 Presenting Structure

Tables of contents are one means with which textbooks traditionally give a global view of their hierarchical structure. These also provide direct access to particular sections with page numbers, which function effectively as links. Presenting hierarchical groups often involves adding introductory sections around a group's subsections. Here we present adaptations of these techniques for presenting generated structure.

Conveying Hierarchy. Systems should properly communicate hierarchical structure so that the user understands the relationships between the search returns that this structure represents. Hierarchical list displays are a commonly used means of conveying such hierarchies with spatial layout. Folding such displays lets users more quickly navigate such structure, which is particularly useful for large hierarchies. Quick navigation of traditional search engine results lets users overcome the inaccuracies inherent in automated search because user's can quickly check the links and choose those that match. We have found that this principle works for hierarchies as well as flat lists of search results. That is, quick navigation of hierarchies helps users work around the inaccuracies inherent in their automated generation. These means of communicate the hierarchy as a whole helps make the global interface itself a unified document about the given topic.

Introductions. Documents do not just place their components in hierarchical groups — they also describe the nature of these groups. Text books typically do this with introductions to sections. An introduction describes what is true for a section as a whole before going into the details of its components, thus helping communicate to the user what significance the section itself has and what relationships exist between its contents. The derivation of document hierarchy from RDF that this paper describes assigns each group an RDF component representing that group's commonality. The local interface can display this grouping component in the same manner as it would the search matches. While the Topia demonstrator formed groups, it only provided navigation to the original search matches, not to any display representing whole groups [17]. To address this, Noadster presents screen displays for groups as well. Specifically, for each group, Noadster generates a screen display whose focal point is the resource URI for the value in the property-value 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 introduction subsections in text books, as compared to introductions consisting of a few paragraphs with a header.

3.4 Adjusting Structure

As with traditional web search engine results, the global interface we propose here for the Semantic Web is not the end result but the start of the journey for the user. As with the document web, our global interface provides direct links to information potentially relevant to the user's need. However, the user can also navigate the document structure systems such as Noadster creates around these returns. We describe both these types of navigation here.

Property Weights. In a given domain, and for a given user, some concepts are more important than others for generating document structure. Therefore 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 [17]. Here, users specify weights of significance for a selection of RDF properties from the ARIA RDF. This allows the concept lattice algorithm to recognize smaller clusters as significant enough to form hierarchical groups if the properties that form them are more significant than competing larger groups sharing less important properties. Similarly, smaller groups can appear sequentially before larger groups. Since Topia gives all

properties a default middle score, users do not need to specify style to start accessing a new RDF repository. As they grow familiar with a repository, they can incrementally adjust the scores of one or more properties.

Domain-independent Property Weights. While Topia's list of weighted properties is hard coded and thus domain-specific, a system could easily generate a list of all RDF resources used as properties and place it in the same type of interface. One potential problem is that repositories with many properties can generate lists that are too long for users to manage. However, as in Topia, giving all properties a default middle score lets this domain-independent adaptation generate default document structure initially and then allow users to incrementally improve the clustering style.

Beyond Concept Lattices. Concept lattices are just one of many potential clustering techniques an RDF style sheet designer can use 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 numeric 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 Noadster uses for global structure. Therefore, integrated global and local system like Noadster can integrate any of these structuring techniques into the rendering of their global and local interfaces.

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. The Samp*L*e system offers such increased user involvement in altering automatically selected content and generated structure [7]. Samp*L*e works for a specific RDF repository, thus inviting integration with this paper's domain-independent foundation.

4. LOCAL INTERFACE

Having described how to get a document structure of relevant links from an RDF repository, we now describe how to display each link when selected. As with typical WWW search, this Semantic Web-based approached renders displays for particular URI's. However, what makes up a display for a URI is different for the Semantic Web than for the traditional web. For the traditional display, the URI locates an existing document. This paper, on the other hand, treats a URI as a starting point for generating a new display. The local interface presents information regarding a single component in the repository. It 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.

4.1 Selection

Selection via search puts the end-user in a position similar to the author. While the typical end-user task, as described in the previous section, is to find existing complete documents that fulfill the current information need, the author uses search to find the raw materials for putting into a new document. This subsection discusses the selection of media to display for one specific subject in an RDF repository.

Focal Point. 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 statements spanning out from it that collectively provide information. The focal point

itself is devoid of content. The media conveying it comes from its statements. The system thus selects media content from these statements.

Associated Statements. Since the statement is the repository's basic unit, there must be at least one statement associated with the current focal point. We identify the *associated statements* as the set of statements including the focal point as either subject, value or property. While viewing the current focal point, the user has direct access to all resources sharing a statement with the focal point.

Literals. While the RDF repository is mostly a collection of statements using URIs as nodes and edges of the graph, some statements involve literal values. Since these consist of plain text, they are much better suited for direct display in the presentation than URIs. As shown in Figure 4, 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" [19]. As part of RDFS, this property thus has a semantic significance that that applies to all RDF(S) repositories, making it essentially domain-independent. That `rdfs:label` is a component's name makes it an especially useful and important piece of text as well.

Displaying Labels. Sesame's explore mode 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 4, instead of as an entry in the main display. Finally, Noadster gives each entry in the global view its label, if it has one (otherwise, it shows its URI). Therefore, 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.

Comments. Another presentable RDFS construct is the `rdfs:comment`. RDFS defines this as "a human-readable description of a resource" [19]. The comment has the same domain-independence as `rdfs:label`. Noadster gives RDFS comments a special display just under the title at the top of the local view, instead of with the other main display entries, as shown in Figure 4.

Inferencing. Inferring label and comment properties from domain-specific ones is an efficient way to make repositories more accessible to generalized RDF(S) browsers. Authors can encode such inference by making certain domain-specific properties a `rdfs:subProperty` of either `rdfs:label` or `rdfs:comment`. This requires only one RDFS triple for each property to make all of its instances labels or comments. In the ARIA repository, for example, the names of artists are made subproperties of `rdfs:label`. With this single RDFS assignment, all artist names become recognized and displayed as labels.

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 images. When a focal point shares a statement with an image resource, Noadster presents it directly in its display. This shows the user both that there is an image and what that image is. Noadster applies a number of standard strategies to find out the MIME type of a resource for improving the display. In Figure 4, for example, the image resources related to the painting appear directly in the associated statements.

4.2 Structure

Allowing access to all the URIs related directly to the current fo-

cal point provides a user interface challenge if there are many statements associated with the focal point. The layout should group the statements 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.

RDF(S)-based Grouping. Noadster groups statements by shared subject, property or value. This method is based on distinctions that can be seen in the "flat" RDF graph structure defined by directly encoded statements. Additional spatial grouping can come from components joined by the `rdfs:subClass` and `rdfs:subProperty` properties.

Clustering Statements. The global clustering techniques we presented 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 property-value in statements for which multiple resources each act as one of the statement's subject. In Noadster's case, we already mentioned common single statement roles as an important clustering characteristic. Other potential characteristics include the namespace of the subjects and values and the role the focal point plays in the statements. Such clustering determines the grouping strategy that puts most related statements 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 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 statements than can appear in a single display. A potential solution is applying clustering techniques to group statements 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.

5. INTEGRATED INTERFACE

Given the generation of well-organized displays as described in the previous sections, the next consideration is where to go next. Both global and local interfaces provide links navigating to new displays. This section describes how to coordinate the navigation both provide.

Full Repository Access. 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.

Basis for Selection. Often the local display results from the user clicking on an entry in the global interface. In this case, the current focal point represents a match of the original request. Traversing links through the local interface can also display focal points matching the original case. In either case, when displaying such matches, it helps the user to show that the node matches the request and why. Noadster does this by highlighting the matching string in the display of the relevant literal.

Lost in Semantic Space. Each statement involving the focal point offers two hyperlink destinations. This often gives the user overwhelming number of choices in local displays. Furthermore, the local interface enables navigation through the entire RDF repos-

itory, offering an overwhelming number of potential current locations. Given all this, users can become easily lost. Therefore, users need to understand where they are in relation to the rest of the repository [13]. While systems should provide full navigation to let users go anywhere in the repository, some guidance and orientation is essential for helping the user make appropriate choices. We describe next two techniques for doing so: showing the local position in the global structure and highlighting local neighbors that also appear in the global structure.

Current Location. Coordination between the local and global interfaces provides such guidance. One important coordinating device is indicating which global interface entry is the current focus when the current location is in the match list. Noadster conveys this by highlighting the entry in the global interface that corresponds with the current local display.

Showing Cross-References. Sometimes the global and local interfaces have links in common. These are analogous to cross-references from printed text documents. They represent relationships between the current location and locations elsewhere in the hierarchical structure generated. Systems should signal these links to the user with distinctive style in both views, as does Noadster. While links from the local interface can lead the user away from the selected information, these links show what information in the local display is relevant to other parts of the generated document.

6. CLOSING

This paper describes creating meaningful presentations from RDF-annotated media repositories. This opens up the Semantic Web as a whole to immediate access by any user using one system. It also serves as a foundation for "semantic style" that content providers can specify per domain and users can specify per domain as well as for RDF-derived displays in general.

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 generate 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 access to RDF, the logical next step is exploring its extension into different document genres, keeping the domain-independent functions as a common foundation for all domains while facilitating development of the domain-specific aspects of each. Noadster takes this approach. The XSLT code defining Noadster allows inclusion of external XSLT files defining presentation for specific domains, starting with Dublin Core. This plug-in methodology adds these domain-specific sub-displays to the main display for each node, generating 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 on search engines as retrieving content instead of documents makes them on-demand generators of new presentations rather than retrievers of existing ones. The key difference between search engines and presentation generation is the granularity of their components. Search engines typically return entire documents, which have multiple components and internal structure. Hypermedia presentation generation, on the other hand, typically handles individual media objects and small clips of text. This finer granularity greatly liberates the possibilities for document generation 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 might suggest 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 those a human expert librarian would return for a document request, they have nonetheless become the main entrance to the WWW. This is because 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 by taking simple assumptions and a simple model and processing them in bulk generates enough sense to help.

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

Acknowledgments

7. 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] 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.

[6] 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.

[7] 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.

[8] C. Fluit, M. Sabou, and F. van Harmelen. Supporting User Tasks through Visualisation of Light-weight Ontologies. In S. Staab and R. Studer, editors, *Handbook on Ontologies in Information Systems*, pages 415–434. Springer-Verlag, 2003.

[9] B. Ganter and R. Wille. Applied Lattice Theory: Formal Concept Analysis, 1997. preprint.

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

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

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

[17] 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.

[18] 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.

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