

Towards Ontology-Driven Discourse: From Semantic Graphs to Multimedia Presentations

Joost Geurts, Stefano Bocconi, Jacco van Ossenbruggen, and Lynda Hardman

CWI, P.O. Box 94079, 1090 GB Amsterdam, The Netherlands,
{Firstname.Lastname}@cwi.nl

Abstract. Traditionally, research in applying Semantic Web technology to multimedia information systems has focused on using annotations and ontologies to improve the retrieval process. This paper concentrates on improving the presentation of the retrieval results.

First, our approach uses ontological domain knowledge to select and organize the content relevant to the topic the user is interested in. Domain ontologies are valuable in the presentation generation process, because effective presentations are those that succeed in conveying the relevant domain semantics to the user. Explicit discourse and narrative knowledge allows selection of appropriate presentation genres and creation of narrative structures, which are used for conveying these domain relations.

In addition, knowledge of graphic design and media characteristics is essential to transform abstract presentation structures into real multimedia presentations. Design knowledge determines how the semantics and presentation structure are expressed in the multimedia presentation. In traditional Web environments, this type of design knowledge remains implicit, hidden in style sheets and other document transformation code. Our second use of Semantic Web technology is to model design knowledge explicitly, and to enable it to drive the transformations needed to turn annotated media items into structured presentations.

1 Introduction

When applying Semantic Web technology to data-driven Web sites, the typical focus is either on annotation of the data to improve information retrieval, or on structuring the data, using Semantic Web modeling techniques as an alternative to database-oriented (including ER) or software-oriented (including UML) modeling techniques [7,9,10,13]. The presentation of the retrieved and structured data is seen as a “detail” that is best left to CSS or XSLT style sheets. We claim that this approach grossly underestimates the importance and complexity of effective presentation design.

To do their work properly, we expect human professional designers to at least understand:

1. the underlying semantics of the client’s information;
2. the most effective order, grouping and priorities for structuring this information;
3. the most effective means of using the chosen medium to convey the information.

Information presentation design is thus an inherently knowledge-driven process. It also requires sufficient knowledge about the domain to be able to convey the essential semantic relations in the presentation. It requires knowledge about how to order, group and prioritize this information effectively, for example by organizing the information into a coherent storyline with a clear introduction, main body and conclusion. Finally, it requires knowledge about media design. First, the designer needs to be able to select the most appropriate medium and, second, to understand the characteristics of that medium to find the most effective means of using its characteristics to achieve the communication goal.

Unfortunately, deploying professional designers to design data-driven Web sites is only feasible if the underlying data, its semantics and target audience are relatively homogeneous. The variety of data sources and semantic relations, combined with a variety of output devices and different user profiles, quickly leads to a combinatorial explosion that forces content providers into a one-size-fits-all approach that ignores the different types of knowledge sketched above. Clearly, some form of automation is needed, and this automated process will need to take this knowledge into account.

While we do not claim that we can even approach the quality of human designers, this paper explores to what extent Semantic Web technology can be used to model the knowledge sketched above, and to what extent this can be used to create an ontology-driven transformation process that generates a coherent multimedia presentation which conveys the relationships within the information to the end user. This paper is explicitly *not* about new models of genre, discourse or narrative. The discourse models presented are used as an example and are inspired by existing communication modeling techniques [11]. The same applies to the document and graphic design models used, which were inspired by general text books on design [19,27].

We assume that the multimedia items of our presentation are properly annotated, and that these annotations represent the domain relations between the items in some sort of semantic graph (e.g. in RDF). We then use this graph, and the associated ontology, to select, order, group and prioritize the information. In addition, we use a *Discourse* ontology to guide this process. This ontology contains information about different document genres and the building blocks for creating documents for each genre. The resulting data structure is, following Rutledge et al. [20], called a *structured progression*. This data structure is then used as the basis for generating a multimedia presentation. In addition to discourse knowledge, this process is also guided by media design knowledge from the *Design* ontology.

Figure 1 illustrates our approach, emphasizing the two focal points of this paper: (1) the process that uses domain and discourse knowledge to transform a semantic graph into a structured progression, and (2), the process that uses discourse and design knowledge to transform this into a multimedia presentation.

The remainder of this paper is structured as follows. Section 2 presents a working example, based on a semantic graph modeling the life and works of the painter Rembrandt van Rijn. This graph is used as a basis to generate a multimedia biography automatically. We use the example to illustrate the processing steps involved and which knowledge sources are used when. Section 3 describes the first process in more depth. We show how domain relations can be mapped to discourse structures, a technique which makes

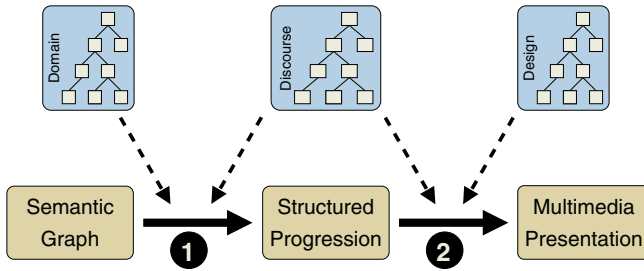


Fig. 1. The two-step, ontology-driven transformation process

our approach extensible to other domains. In addition, we report on our experience using existing Semantic Web technologies. In Section 4 we describe the second process, which transforms a structured progression to a real multimedia presentation. This step is based on transformation rules whose application is guided by the discourse and design ontologies, and also by the constraints on the final multimedia presentation. The rules use the grouping, ordering and priorities of the material defined in the structured progression to create a coherent multimedia presentation. In Section 5, we describe related work and examine the design choices of other systems with respect to the main points of our approach. Finally, conclusions and future work are given in Section 6.

2 Example Scenario

Our user queries for “life and work of Rembrandt” using a Web browser. The user has also selected the preferred types of the structured progression. This type will define the *genre* of the presentation. Examples of genre include `disc:Biography` and `disc:CV`¹. Our user has chosen `disc:Biography` among the preferred genres. Finally, the user selects the output medium, e.g. printed paper, interactive hypermedia presentation, non-interactive multimedia, a slide show, etc. In our case the user has chosen a non-interactive multimedia presentation.

The results returned by the retrieval component (outside the scope of this article), combined with the associated semantic relations from the domain ontology, form the semantic graph that is the main input to our system (shown at the left of Figure 2). Based on this graph, the system selects a matching genre type from the ones specified by the user, in our case the `disc:Biography`. It then uses a set of rules to transform the graph into a structured progression representing the biography. At this stage, the information in the semantic graph has been ordered, grouped and prioritized (center of Figure 2).

The structured progression contains some typical facts about Rembrandt, such as his full name, place and date of birth, etc. This is followed by information about his career and finally some information about his personal life, that is, about his son Titus and wife Saskia. In the next step, media items which represent the concepts are retrieved. The

¹ Schema level information is typeset in sans-serif font, with a namespace-like qualifier to indicate the ontology used. So `disc:Biography` refers to the Biography class of the discourse ontology, and `dom:Artist` to the Artist class of the domain ontology.

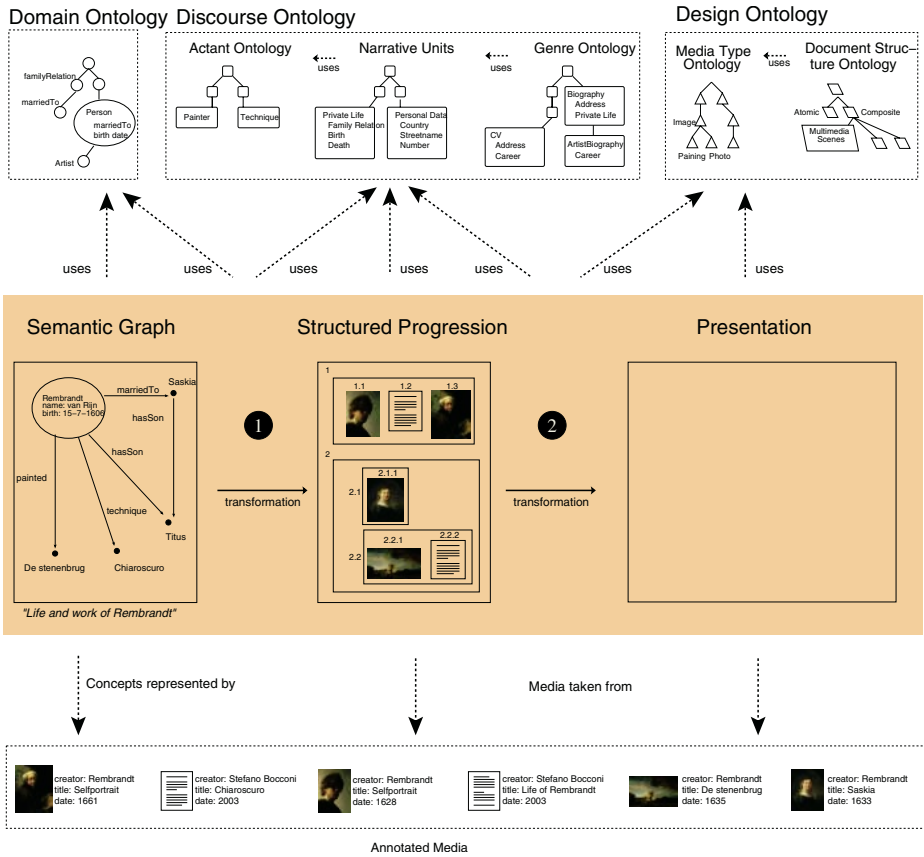


Fig. 2. Sketch of the processing of the example scenario. The middle layer shows an example semantic graph (left) that is first transformed to a biography structure (middle) and then to a multimedia presentation (right). The upper layer shows the ontologies involved, while the lower layer represents the annotated media items used.

chosen output medium defines how the biography is conveyed to the user. A multimedia presentation, for example, will use sequentially ordered scenes and possibly sub-scenes. Transitions between scenes can be realized by using links or by placing them one after the other using appropriate fade in/out effects. In our case the latter is used. Finally the user chooses a format for the presentation. For the timed multimedia presentation generated, the system supports SMIL and HTML+TIME. The presentation is serialized using the selected output format and sent to the user's Web client (on the right of Figure 2).

3 From Semantic Graph to Structured Progression

This section discusses the first transformation step. In terms of the example, this is the step that transforms the semantic (RDF) graph representing the relevant media items and their annotations into a target biography structured progression.

Note that traditional document processing chains usually *start* at this point. For example, one could apply an XSLT transformation or CSS style sheet to the biography structure to produce presentation formats such as HTML. These methods are, however, not well suited to generate RDF and XML structures from RDF graphs. When applied to RDF, languages such as CSS and XSLT operate purely on the XML level of RDF's serialization syntax, without any understanding or support for the semantics of the RDF data model. Since domain knowledge is essential for conveying the underlying message, the transformation process also needs access to knowledge on the RDF Schema level to be able to query the underlying domain ontology. In addition, since the discourse knowledge is also stored in an RDF Schema ontology, RDF Schema support is also required to gain access to the transformation's own "operating" knowledge.

We have prototyped several transformations in both Java and Prolog environments that have direct access to a Sesame RDF Schema-based repository [5,6]. We use a Sesame server to store all RDF and RDF Schema knowledge used. The key advantage of this approach is that our transformation no longer needs to access the RDF on the XML syntax level, but can gain direct access on the RDF instance level and RDF Schema level using any of the query languages supported by Sesame (currently RQL [14], RDQL [18] and SeRQL [1]).

This approach allows us to make all domain and discourse knowledge explicit that usually remains implicit in the XSLT (or, in our case, Java or Prolog) transformation code, by modeling it in the appropriate ontology. As a consequence, the transformation process itself uses (declarative) domain and discourse-specific knowledge, while the (procedural) transformation code remains generic.

For example, given the instance Rembrandt², from the semantic graph on the left of Figure 2 on page 600, the transformation code uses an RQL query to retrieve the classes this instance belongs to, and selects a structured progression type from the discourse ontology that can be applied to this class. In our example, Rembrandt turns out to be an instance of `dom:Artist`, and the discourse ontology defines an instance of `disc:ArtistBiography` that has a `disc:Subject` property with value `dom:Artist`.

Each structured progression, including `disc:ArtistBiography`, has a `disc:narrativeUnits` property that specifies the `disc:NarrativeUnits` that can be used to construct the structured progression. In our example, the applicable narrative units are instances of the classes `disc:Personal Data`, `disc:Private Life` and `disc:Career`.

Narrative units have associated rules which are inspired by the work of Greimas [11] (see Table 1 on page 602) and are used to select matching content. An instance of `disc:Private Life`, for example, contains rules to select information about family relations from the semantic graph. In our case, the semantic graph includes an `dom:isMarried` relation between Rembrandt and Saskia.Uylenburgh. Rule # 3 in the table, for example, can use this domain relation to select Saskia in the `disc:Role` of `disc:Spouse`.

² Instance level information is typeset in courier font.

Table 1. Narrative units and their associated rules (simplified for readability)

PersonalData					
#	Current Role	Relation	Progression	New Role	Relevant Properties
1	MainCharacter	none	none	none	[descriptionText, portrait, ...]
...					
PrivateLife					
#	Current Role	Relation	Progression	New Role	Relevant Properties
2	MainCharacter	none	none	none	[name, birthDate, ...]
3	MainCharacter	isMarried	privateLife	spouse	[marriedPlace, marriedDate]
4	MainCharacter	parentOf	none	child	[birthPlace, birthDate]
5	spouse	none	none	none	[name, daughterOf]
...					
Career					
#	Current Role	Relation	Progression	New Role	Relevant Properties
6	MainCharacter	none	none	none	[workPlace, ...]
7	MainCharacter	taughtBy	career	master	[stylePeriod]
8	master	style	career	technique	[styleDescription]
...					

```

<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:="http://www.cwi.nl/~media/ns/discourse#"
  xmlns:d="http://www.cwi.nl/~media/ns/discourse#"
  >
  <discourse d:type='biography'>
    <narrativeunit rdf:parseType='Resource'>
      <type>PersonalData</type>
      <actant>Rembrandt_van_Rijn</actant>
      <role>MainCharacter</role>
      <data rdf:parseType='Resource'>
        <descriptiveText>Rembrandt Harmensz. van Rijn is the most famous...</descriptiveText>
        <portrait rdf:resource='http://www.rijksmuseum.nl/SK-A-4691.Z.jpg'>
          </data>
      </narrativeunit>
    <narrativeunit rdf:parseType='Resource'>
      <type>Career</type>
      <actant>Rembrandt_van_Rijn</actant>
      <role>Painter</role>
      <data>...</data>
    <narrativeunit rdf:parseType='Resource'>
      <type>Career</type>
      <actant>Chiaroscuro</actant>
      <role>Technique</role>
      <data rdf:parseType='Resource'>
        <descriptiveText>Clair-obscur (French) and chiaroscuro (Italian)...</descriptiveText>
        <painting rdf:resource='http://www.rijksmuseum.nl/SK-A-1935.ORG.jpg'>
          <painting rdf:resource='http://www.rijksmuseum.nl/SK-A-4691.Z.jpg'>
            </data>
          </narrativeunit>
        </narrativeunit>
      <narrativeunit rdf:parseType='Resource'>
        <type>PrivateLife</type>
        <actant>Rembrandt_van_Rijn</actant>
        <role>PrivatePerson</role>
        <data>...</data>
      <narrativeunit rdf:parseType='Resource'>
        <type>PrivateLife</type>
        <actant>Saskia_Uylenburgh</actant>
        <role>Spouse</role>
        <data rdf:parseType='Resource'>
          <descriptiveText>In 1634 Rembrandt married...</descriptiveText>
          <portrait rdf:resource='http://www.rijksmuseum.nl/SK-A-4057.jpg'>
            </data>
          </narrativeunit>
        </narrativeunit>
      </discourse>
    </rdf:RDF>
  
```

Fig. 3. RDF/XML representation of the biography

Selecting roles is the main task of the rules each narrative unit contains. Each rule states that if a given instance with a given role (e.g. Rembrandt in the role of `disc:MainCharacter`) has a certain domain relation with another instance (e.g. relation `dom:isMarried` with `Saskia_Uylenburgh`), the latter instance could become a new character in the story with the role specified in the rule (in this case Saskia enters the story in the role of `disc:Spouse`).

Rules can be applied recursively to define how to further expand the story with the new character. In our example, rule # 3 not only assigns to `Saskia_Uylenburgh` the role of `disc:Spouse`, but it also specifies that `PrivateLife` is the narrative unit that can be used for a subsequent nested story line. The engine would then look whether `PrivateLife` contains rules that can be applied to `disc:Spouse`. If there is such an applicable rule, a nested narrative unit about e.g. the son of `Saskia_Uylenburgh` would be added to the presentation. This process continues until no more rules can be applied or no more elements satisfy the current rule. A rule can also specify that no further expansion should happen.

Note that since each narrative unit uses different rules and different roles, the same instance can play a different role in the story fragments generated. For example, `Saskia_Uylenburgh` with role `disc:MainCharacter` would generate a different story from `Saskia_Uylenburgh` in the role of `disc:Spouse`.

A rule also defines, for each specific role, the information from the domain ontology that needs to be incorporated in the narrative unit. For example, rule # 1 specifies that for the `disc:MainCharacter` in `disc:PersonalData` extra slots should be added: `disc:descriptionText` and `disc:portrait`. The rules also define (not shown in the table) which properties from the domain ontology contain the corresponding information. Rules, as well as all the other concepts used in the structured progression, are explicitly encoded in the discourse ontology.

After all rules have been applied to the initial arbitrary semantic graph structure, the result is the structured progression represented graphically at the center of Figure 2 on page 600 and in RDF in Figure 3. This is a biography with three narrative units: `disc:PersonalData` with Rembrandt in the role of `disc:MainCharacter` with two other units, `disc:Career` containing `Chiaroscuro` in the role of `disc:Technique` and `disc:PrivateLife` containing `Saskia_Uylenburgh` in the role of `disc:Spouse`. The information contained within the initial semantic graph has been grouped, ordered and prioritized in terms of the `disc:Biography` discourse. This hides the specific details of the domain ontology from the next phase of the transformation. The structured progression, therefore, has to pass on sufficient semantics for the next step which transforms the structure progression into a playable multimedia presentation.

4 From Structured Progression to Multimedia Presentation

This section discusses the second transformation step shown in Figure 1 and 2. In terms of the example, this transforms the biography in Figure 3 about Rembrandt into a final form presentation. Although the structure of the information is known at this point, by what means it is conveyed to the user is still open. In this section we show how both

discourse and design knowledge are used to transform the structured progression into a final form multimedia presentation.

First, the structured progression is transformed into a so-called *document structure*. In this step, decisions are made about the output medium (e.g. traditional text, interactive hypermedia or “passive” multimedia³). For traditional text, this step would map the biography-specific elements to a more generic document structure, defined in terms of chapters, sections etc. similar to the structure in \LaTeX or HTML documents. Similar structures can be defined for other output media, such as the multimedia presentation used in the example, that are defined in terms of scenes and sub-scenes.

Second, the document structure is transformed into a tree of formatting objects. In this step, the more detailed layout and formatting decisions are made. For traditional text, this would involve the decisions related to typesetting normally made by word processors or typesetters such as \TeX . For the multimedia presentation, this also includes determining the exact timing of the presentation, the interaction style, transition effects etc.

Note that the advantages of this two-step process are similar in both the text and multimedia cases. By mapping the discourse-specific narrative units to more general document elements, the second step can be defined in more commonly applicable formatting rules. In the text case, this means that we do not have to define the precise formatting of the `disc:PrivateLife` unit. Instead, it can be mapped onto the document section element, relying on the common formatting rules for section-level elements. Similar advantages apply to the multimedia case. For example, instead of specifying explicitly what transition effects have to be applied between `disc:PrivateLife`, `disc:Career` and all the other units, we prefer to specify this once on the more generic scene and sub-scene level, after the narrative units have been mapped to these multimedia document structures.

Unfortunately, the disadvantages of this approach are also similar. The main drawback is that when one looks in more detail, there always comes a level that can no longer be effectively specified in terms of the document structure. In the case of text, this article for example, the section, subsection and paragraph level can be conveniently expressed in \LaTeX . Even the content and structure of the tables can be effectively represented, because \LaTeX defines a sufficiently rich document structure. Figure 2, however, has a level of detail that can no longer be conveniently expressed in \LaTeX . Instead, it was created in an external drawing tool and included as a PostScript figure. As a result, from the perspective of the document structure, the figure is a black box. All semantics of the figure remain implicit, and all formatting within that black box has been done manually. For the multimedia case, the situation is even worse. As a relatively new output medium, multimedia documents have not yet evolved a common structure as rich as textual documents. When zooming into a multimedia document, one thus sooner reaches the moment where one can no longer conveniently represent the intended semantics in the document structure.

In practice, one reaches this point so often, that the option of leaving the formatting of material that is too detailed for the document structure to a human designer is not even feasible. Instead, in the first step the detailed structures are copied directly into the document structure, so we can then define specific rules in the second step to deal with their formatting directly.

³ At the time of writing, our prototype supports only multimedia output.

Table 2. Example mapping from narrative units to document structure.

Biography					
Narrative unit				Document structure	
	Type	Role		Paper	Multimedia
1	PersonalData	MainCharacter		Section	Scene
2	PrivateLife	MainCharacter		Section	Scene
3	Career	MainCharacter		Section	Scene
2.1	PrivateLife	Spouse		SubSection	Sub-scene
3.1	Career	Master		SubSection	Sub-scene
2.2	PrivateLife	Child		SubSection	Sub-scene
3.2	Career	Technique		SubSection	Sub-scene
...

4.1 From Structured Progression to Document Structure

As described previously, a structured progression defines grouping, ordering and priorities of disc:narrative unit instances. It describes the main message an author wants to convey, and relates and organizes narrative units contributing relevant information to this message. It structures the information by defining a hierarchical ordering, differentiating between detailed and more general information. Information without presentation, however, remains abstract. A human author uses different mechanisms, depending on the context, to transform this information into a perceivable document. This includes the selection of media items and the choice of a output medium such as “multimedia presentation” or “paper document”. The structure of a narrative, which in our case is modeled as a structured progression, is partly conveyed through the document structure. In traditional text documents, the chapters, sections and paragraphs reflect the underlying narrative structure, where chapters are more general than sections. Multimedia presentations have a similar structure (although less explicit) which we call *scenes* and *sub-scenes*. These are used as generic grouping mechanisms, for which appropriate formatting rules can be defined. One of the main goals of these rules is to identify the (sub)scene boundaries by using, for example, transition-effects or hyperlinks. The screendump on the right of Figure 2 on page 600 represents the first scene; the thumbnail images in the corner are hyperlinks to the two other scenes.

Table 2 shows an example mapping of discourse structures to appropriate document structures. For example, the narrative unit ‘PersonalData’ is represented as a ‘section’ in a paper document and as a ‘scene’ in a multimedia presentation.

Because narrative units can generate sub-narrative units, which subsequently can result in an arbitrary number of levels in the structured progression, the mapping in general cannot be made explicit. In addition, the document structure for paper documents (e.g. chapter, section, subsection) tends to be more fine-grained than a document structure for multimedia presentations (e.g. scene, sub-scene). As a consequence, when generating multimedia presentations, one has to partly bypass the document structure more often than for paper documents. Instead, one needs to be able to convey semantic and discourse relations directly by means of spatial and temporal layout. In our examples, the relation between paintings and their titles is too application-specific to be expressed in our generic

document structure. As an escape-mechanism we still allow this information to be copied literally into the multimedia document structure, so that specific formatting rules can determine the formatting of these domain semantics directly.

4.2 Conveying Document Structure

The final step discussed here is the transformation of the document structure into a set of presentation constructs that specify all formatting decisions needed to generate the multimedia presentation. This includes assigning layout clues to allow a reader to recognize the document structure (and thus the underlying semantic relations). For paper documents this is typically realized by typographic conventions, such as using large bold-face fonts for headings to mark the beginning of a new section and chapter. Multimedia presentations can sometimes use similar typographic conventions, but also use more multimedia specific means. Scene boundaries, for example, can be conveyed by using transition effects, or by using hyperlinks to traverse from one scene to the other.

The process of transforming a document structure into presentation constructs uses our Cuypers library (see [23,24] for details). This library uses constraint solving techniques to verify whether a presentation construct conforms to the delivery-context constraints, such as screen size. If these constraints are violated then the library allows us to specify alternative formatting strategies, or indeed discard the construct as a whole. To anticipate formatting failure, the transformation process needs to proceed bottom up, requiring the transformation of leaf nodes of the document structure first.

Because a user may want to be able to distinguish visually between `disc:PrivateLife` and `disc:Career` the transformation needs knowledge about the type of narrative corresponding to the scene. In addition, the role of the characters in a scene also need to be specified: the presentation of a `disc:Son` may need to be different from the presentation of a `disc:Spouse`. A rule which transforms a document structure, corresponding to a narrative unit, into a presentation construct has thus two discourse parameters: `disc:NarrativeType` and `disc:Role`. These discourse parameters allow our system to adapt the formatting of the presentation to convey the message more effectively. Because both parameters are defined in the discourse ontology, we can use subsumption reasoning to generalize the rules, making them applicable for presenting multiple discourse relations. For example, a rule which presents a `disc:Painter` can be generalized to `disc:Artist`, in which case the same rule is also applicable to `disc:Sculptors`.

4.3 Conveying Discourse Semantics Directly

As mentioned before, almost every non-trivial application has some more specific details in the domain or discourse semantics that cannot be effectively expressed in the generic document structure. We allow these semantics to be copied into the document structure. As a result, the formatting step needs to have specific rules that can deal with this type of information, in addition to the rules discussed above that only deal with the document structure proper. We experimented with such specific rules and used different types of knowledge to determine the formatting.

Discourse Knowledge. In our example, `disc:Portrait`, `disc:Painting` and `disc:Descriptive-Text` represent concepts directly by means of a media item, and these concepts are mapped directly onto equivalent concepts in the multimedia design ontology (e.g. `mm:Painting` etc.). Depending on their function in the discourse, we need to define how different media types are to be formatted. For example, the same image of a self-portrait of Rembrandt might require different formatting, one type of formatting when it functions as a `disc:Portrait` in the `disc:PersonalData` section, and another when it functions as an example `disc:Painting` illustrating the *Chiaroscuro* technique. In addition, the system needs knowledge about how to deal with media items that are related in the discourse, and how to group them in terms of presentation constructs that are specific for the output medium. In our example, it will need to be able to group the range of paintings used to illustrate *Chiaroscuro* into a slide show, but it would choose a different solution for a paper version.

Media Characteristics. When building a balanced and coherent presentation, one needs to be aware of the differences between the various media modalities used. Text, images, audio and video all have a very different impact on the way the presentation is perceived by the user, and the system needs to be aware of the differences⁴. In addition, different media types have different formatting requirements. The images of the paintings in our example have a fixed aspect ratio, while the text boxes used for the descriptions can have a wide range of acceptable aspect ratios.

On the other hand, some media types also share important characteristics. For example, the rules mentioned above discriminate between `disc:Portrait` and `disc:Painting`. When these rules fail or are missing, there would still be a fall back rule that defines a generic formatting for images. This rule would be applied because the media type ontology defines both `mm:Painting` and `mm:Portrait` to be subclasses of `mm:Image`. In addition, similar subsumption-based fall back rules are required for the groupings. That is, in the absence of specific rules to group paintings into slide shows, fall back rules specifying the generic formatting of a group of images are required.

In practice, the variety of document and discourse structures, media items and intended layout effects, and all the possible combinations, quickly result in an overwhelming number of rules if each of them has to explicitly specified. By using ontological knowledge, however, we can generalize rules, matching on entire subtrees instead of individual cases. For cases where these rules are too generic, more specific rules can be added later. This approach also provides a redundancy that can be exploited for backtracking. In the case that a rule cannot be applied, because of, for example, hardware constraints (e.g. the resulting presentation would not fit on the screen), an alternative rule is invoked automatically. Only if no rule is applicable does the presentation generation fail (see [23, 24] for details).

Once all levels of document structure have been represented by presentation constructs, a final step is required to generate the final presentation format. This is a relatively simple XSLT transformation. We currently support SMIL 1.0 [25], SMIL 2.0 [26] and HTML+TIME [21].

⁴ Most of this knowledge is, however, still implicit in the current version of the system.

5 Related Work

The common thread among the related work is the use of semantics to generate meaningful presentations. The Standard Reference Model for Intelligent Multimedia Presentations [4] provides a theoretical framework which many of these systems follow. It is, however, the different approaches to realizing the overall goal and the context in which they do this that differs. In discussing related work we focus on the issues of domain independence, explicit knowledge representation and the type of generated narrative.

Bateman et al. [2] describe a system (D*Art_{bio}*) that generates artist biographies based on a domain model containing information about several thousand artists. The presentation is created following presentation plans, which are a type of template expressed in terms of Rhetorical Structure Theory [17]. These plans are genre-specific (e.g. for biographies) and tailored to the information contained in the domain model. A natural language generator and a graphic generator create the information items and a layout component determines the layout.

D*Art_{bio}* is domain dependent and focuses neither on the reuse of information nor on making the knowledge it uses explicit; in this regard it is a pre Semantic Web application, because the knowledge sources are not separate from the core functionality of the system. Furthermore, the narrative it can generate is based on templates (presentation plans) and, once the genre has been determined, has a fixed structure.

Artequakt [15] also generates artist biographies. The system uses a biography ontology, which defines the data for an artist biography. Information is collected by parsing text found on the Web and is subsequently presented using templates. Artequakt does not use annotated media items and, because of the parsing, is strongly text oriented. This allows, at least in theory, the reuse of all the textual information published on the Web about artists. On the other hand, Artequakt uses a domain dependent ontology as its knowledge base and domain dependent templates, which would need to be recreated to apply the system to other domains.

Rutledge et al. [20] describe the Topia system, which generates hypermedia presentations in a domain-independent way, using clustering of annotated media items. The clustering is made by grouping media items that have the same attributes with the same values. Topia does not use knowledge about a particular domain or a particular attribute because it only needs to know whether two attributes or values are equal or different. The advantage of the Topia approach is that the grouping algorithm is relatively simple and, more importantly, that it is domain independent. This is at the same time the limiting factor, because domain knowledge can contribute to the understanding of a topic since the frequency of an occurring relation does not necessarily indicate its importance and might not answer the question a user has. Topia has no explicit discourse knowledge, so that it cannot generate an evolving narrative such as, e.g. a biography or a fairy tale. Topia allows some flexibility because the user can indicate the most important relations, thus guiding the clustering algorithm to cluster media items on these relations. In this sense, we might say that Topia makes the knowledge and the reasoning it uses accessible to the user.

Little et al. [16] generate presentations using media items annotated with Dublin Core [8] metadata. Their focus is on inferring knowledge from the metadata, thereby creating and extending the semantic graph. Dublin Core has the advantage that large

amounts of content have been annotated with it; moreover, because Dublin Core is generally applicable, the generation process has a certain degree of domain independence. The disadvantage is that Dublin Core is not a powerful standard to express semantics. Therefore only simple narratives can be generated, by using grammars to define which semantic relations can be put in an introduction, in a middle or ending section.

Some analogies exist between our work and Scholonto [22], even if the latter is not meant for presentation generation. ScholOnto is designed to enable researchers to describe via a semantic network the contributions a document makes and its relationship to the literature. They also use a discourse ontology, in their case to model at a general level the claims a researcher would want to make about a document, while our focus is more on narrative and on the single media item level. ScholOnto is domain specific, its domain being scientific documents, and requires documents to be annotated with the discourse ontology (in ScholOnto there is no domain ontology, or the domain ontology and discourse ontology are the same).

6 Conclusions and Future Work

This paper explores the use of Semantic Web technology for explicit domain, discourse and media knowledge for conveying domain and discourse relations in multimedia presentations. Including these different types of knowledge into the process explicitly also allows for the adaptability and extensibility required.

To date, we have generated only short presentations, based on a restricted domain ontology. We also focused on a single discourse structure (the biography) and a single document structure (a multimedia presentation). More research is needed to scale these aspects of the system to more realistic scenarios. We are also investigating the possibilities of taking explicit user profiles into account, and how knowledge about the user interacts with the discourse and design knowledge used in the current prototype.

We have focused on two characteristics that are important for a presentation generation system: 1) to be effective in conveying the relevant domain semantics, and 2) to be generally applicable. These two requirements are often conflicting. In order to present relevant domain relations but keep the process as general as possible, we query the domain ontology only when transforming the semantic graph into a structured progression. All important domain relations are then mapped to discourse relations. This explicit mapping localizes all specific domain knowledge in the first step. This has the advantage that the application in the remaining transformation always deals with known discourse concepts and is therefore reusable for different domain ontologies.

Our approach to explicit discourse knowledge uses rules with explicit roles to dynamically expand a story. This has the advantage that information presented can be adapted to the context in which it is presented, and helps to improve the coherence of the overall presentation. Using role-based rules can lead to narrative complexities such as recursive expansion of narrative units which need to be dealt with by the designer of the rules.

Another design goal was to strive for making all reasoning explicit, with the assumption this facilitates information sharing and flexibility. All the intelligence of the engine creating the presentation is RDFS-encoded. The discourse ontology not only allows our

application knowledge to be made explicit, but is also used as a logical configuration tool (and also graphical, if using a graphical ontology editor like Protege-2000 [12])⁵. This is because all elements determining the behavior of the application are contained in the ontology and are defined in terms of each other: a genre is defined by its narrative units, which are defined by their rules, which are expressed in terms of roles. The ontology defines thus the framework a narrative designer would use to define his or her particular form of narrative.

The fact that an ontology editor can be used to configure the application is a secondary but interesting application of Semantic Web technology. The drawback of this approach is that ontology languages such as RDF Schema are not designed for expressing rules. As a result, our rules are forced to be simple. One cannot combine rules using logical AND or OR or make one rule dependent on the outcome of another. A next step is to investigate the use of more powerful languages such as RuleML [3] for expressing the rules within the system.

The second part of the system uses discourse knowledge to convey appropriate semantics for presenting information. Conveying semantics in terms of presentation is partly realized by document structure which materializes the structure of the information. The way this structure is made explicit changes for different types of document structure and modalities. Especially for multimedia presentations, document structure by itself is not sufficient to convey relevant semantics. This is mainly due to the fact that multimedia presentations, in contrast to paper documents, have a shallow document structure. To compensate for this, our system may bypass the document structure and use lay-out rules that operate directly on the discourse semantics.

The transformation of the structured progression to a final form presentation is carried out using two types of rules — those that allow a generic approach ensuring that some sort of presentation can be generated; and discourse specific rules, which convey important semantical relations, that can be added easily to the system. This contrasts to the approach in Artequakt and DArt_{bio} where the domain dependent rules are deeply embedded within the system.

Acknowledgments. Part of the research described here was funded by the Dutch national NWO/NASH, ToKeN2000 I²RP projects and by grant IST-2000-28767 from the European Union's Information Society Programme to the Question How project. Examples are taken from a ToKeN2000 demonstrator, and all media content has been kindly provided by the Rijksmuseum in Amsterdam. We like to thank Lloyd Rutledge, Frank Nack, Geert-Jan Houben (TU/e) and Martin Alberink (Telematica Instituut) for their useful feedback.

References

1. Administrator Nederland B.V. *SeRQL user manual*, April 4, 2003.
<http://sesame.aidministrator.nl/publications/SeRQL%20user%20manual.pdf>

⁵ Protege screenshots, the RDFS ontologies used, the online Sesame repositories, together with a demo, can be found at: <http://www.cwi.nl/~media/conferences/ISWC2003/>.

2. J. Bateman, J. Klein, T. Kamps, and K. Reichenberger. Towards Constructive Text, Diagram, and Layout Generation for Information Presentation. *Computational Linguistics*, 27(3):409–449, September 2001. <http://acl.lidc.upenn.edu/J/J01/>.
3. H. Boley, S. Tabet, and G. Wagner. Design Rationale of RuleML: A Markup Language for Semantic Web Rules. In *Semantic Web Working Symposium (SWWS)*, Stanford University, California, USA, July 30–August 1, 2001. <http://www.semanticweb.org/SWWS/program/full/paper20.pdf>.
4. 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. <http://www.dfki.uni-sb.de/imedia/lidos/papers/csi97/>.
5. J. Broekstra, A. Kampman, and F. van Harmelen. Sesame: An Architecture for Storing and Querying RDF Data and Schema Information. In D. Fensel, J. Hendler, H. Lieberman, and W. Wahlster, editors, *Semantics for the WWW*. MIT Press, 2001.
6. 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. link.springer.de/link/service/series/0558/papers/2342/23420054.pdf.
7. S. Comai and P. Fraternali. A semantic model for specifying data-intensive Web applications using WebML. In *Semantic Web Working Symposium (SWWS)*, Stanford University, California, USA, July 30–August 1, 2001. <http://www.semanticweb.org/SWWS/program/full/paper19.pdf>.
8. Dublin Core Community. Dublin Core Element Set, Version 1.1, 1999. <http://www.dublincore.org/documents/dces/>.
9. K. Falkovych, M. Sabou, and H. Stuckenschmidt. UML for the Semantic Web: Transformation-Based Approaches. In B. Omelayenko and M. Klein, editors, *Knowledge Transformation for the Semantic Web*, pages 92–106. IOS Press, 2003. http://www.cwi.nl/~media/publications/UML_for_SW.pdf.
10. M. F. Fernandez, D. Florescu, J. Kang, A. Y. Levy, and D. Suciu. Overview of Strudel – A Web-Site Management System. *Networking and Information Systems Journal*, 1:115–14, 1998. <http://www.cs.wisc.edu/~jaewoo/pubs/NISJ98.pdf>.
11. J. Greimas. *Structural Semantics: An Attempt at a Method*. Lincoln: University of Nebraska Press, 1983.
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. T. Isakowitz, E. A. Stohr, and P. Balasubramanian. RMM: A Methodology for Structured Hypermedia Design. *Communications of the ACM*, 38(8):34–44, August 1995.
14. G. Karvounarakis, S. Alexaki, V. Christophides, D. Plexousakis, and M. Scholl. RQL: A Declarative Query Language for RDF. In *Proceedings of the Eleventh International World Wide Web Conference (WWW2002)*, pages 592–603, Honolulu, Hawaii, USA, May 2002. ACM Press. <http://139.91.183.30:9090/RDF/publications/www2002/www2002.html>.
15. S. Kim, H. Alani, W. Hall, P. Lewis, D. Millard, N. Shadbolt, and M. Weal. Artequakt: Generating Tailored Biographies with Automatically Annotated Fragments from the Web. Presented at the Semantic Authoring, Annotation and Knowledge Markup (SAAKM) 2002 Workshop at the 15th European Conference on Artificial Intelligence (ECAI 2002), Lyon, France.

16. S. Little, J. Geurts, and J. Hunter. Dynamic Generation of Intelligent Multimedia Presentations through Semantic Inferencing. In *6th European Conference on Research and Advanced Technology for Digital Libraries*, pages 158–189. Springer, September 2002. <http://www.cwi.nl/~media/publications/ecdl2002.pdf>.
17. W. C. Mann, C. M. I. M. Matthiesen, and S. A. Thompson. Rhetorical Structure Theory and Text Analysis. Technical Report ISI/RR-89-242, Information Sciences Institute, University of Southern California, November 1989.
18. B. McBride. Jena: Implementing the RDF Model and Syntax Specification. See <http://www.hp1.hp.com/semweb/>.
19. M. Oldach. *Creativity for Graphic Designers*. North Light Books, Cincinnati, Ohio, 1995.
20. 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*, Nottingham, UK, August 26–30, 2003. <http://www.ht03.org/>. To be published.
21. P. Schmitz, J. Yu, and P. Santangeli. Timed Interactive Multimedia Extensions for HTML (HTML+TIME): Extending SMIL into the Web Browser. W3C Note are available at <http://www.w3.org/TR/1998/NOTE-HTMLplusTIME-19980918>, September 1998.
22. S. B. Shum, E. Motta, and J. Domingue. ScholOnto: an Ontology-Based Digital Library Server for Research Documents and Discourse. *International Journal on Digital Libraries*, 3(3), August/September 2000. <http://kmi.open.ac.uk/projects/scholonto/docs/ScholOnto-IJoDL-2000.pdf>
23. J. van Ossenbruggen, J. Geurts, F. Cornelissen, L. Rutledge, and L. Hardman. Towards Second and Third Generation Web-Based Multimedia. In *The Tenth International World Wide Web Conference*, pages 479–488, Hong Kong, May 1–5, 2001. IW3C2, ACM Press. <http://www10.org/cdrom/papers/423/index.html>.
24. J. van Ossenbruggen, J. Geurts, L. Hardman, and L. Rutledge. Towards a Formatting Vocabulary for Time-based Hypermedia. In *The Twelfth International World Wide Web Conference*, pages 384–393, Budapest, Hungary, May 20–24, 2003. IW3C2, ACM Press. <http://www2003.org/cdrom/papers/refereed/p383/p383-ossenbruggen.html>.
25. W3C. Synchronized Multimedia Integration Language (SMIL) 1.0 Specification. W3C Recommendation, June 15, 1998. <http://www.w3.org/TR/1998/REC-smil>. Edited by Philipp Hoschka.
26. W3C. Synchronized Multimedia Integration Language (SMIL 2.0) Specification. W3C Recommendation, August 7, 2001. <http://www.w3.org/TR/smil20>. Edited by Aaron Cohen.
27. R. Williams. *The Non-Designer's Design Book*. Peachpit Press, 1994.