Media, Politics and the Semantic Web
An Experience Report in Advanced RDF Usage*

Wouter van Atteveldt, Stefan Schlobach, and Frank van Harmelen

Department of Artificial Intelligence
Free University Amsterdam (The Netherlands)
De Boelelaan 1071, 1071 HV Amsterdam
{wva,Frank.van.Harmelen}@cs.vu.nl, schlobac@few.vu.nl

Abstract. The media play an important role in the functioning of our society. This role is extensively studied by Communication Scientists, requiring a systematic analysis of media content. The methods developed in this field utilize complex data models and background knowledge. This data is generally represented ad hoc, making it difficult to analyze, combine and share data sets.

In this paper we present our work on formalizing this representation using RDF(S). We discuss the requirements for a good representation, highlighting a number of non-trivial modeling decisions. We conclude with a description of the resulting system and the benefits for a recent investigation of the 2006 Dutch parliamentary campaign. This case study shows concrete improvements for annotating, querying, and analyzing data, but also indicates a number of aspects that were more difficult to model in RDF(S), contributing to the discussion on modeling with and improving RDF(S) and associated tools.

1 Introduction

The media play an important role in our society. Citizen access to information about world events is almost exclusively mediated, making the press a vital part of our democracy. This underscores the need for the systematic study of the media done by Communication Science [1]. *Relational Content Analysis (RCA)* conducts this systematic analysis by representing news content as a graph linking the relevant actors and issues, which can then be used as input for further analysis [2]. This representation is currently mostly informal forcing answers to complex queries to be composed in a procedural way. Moreover, there are differences in the information captured by various RCA methods and in the vocabulary used for existing data sets. These aspects make it difficult to combine and reuse data.

In this paper we describe a recent case-study on how current Semantic Web technology can help to overcome these limitations. Since 1994, Communication

* The authors would like to thank Mark van Assem for his insightful contributions to the discussions leading to this paper and for his comments on the final version. We also thank the reviewers for their thorough reading and useful suggestions.

E. Franconi, M. Kifer, and W. May (Eds.): ESWC 2007, LNCS 4519, pp. 205–219, 2007.

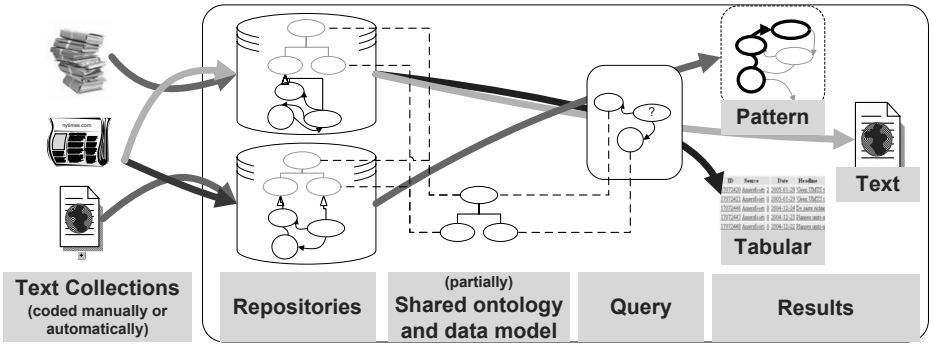

Fig. 1. A Framework for Querying Heterogeneous News Sources

Scientists at the Free University and the University of Amsterdam have conducted an extensive study of the influence of the media coverage of Dutch electoral campaigns on public opinion, most recently in Nov. 2006 [3]. These studies were based on the *NET method*, an RCA framework introduced in [4].

As visualized in Fig. 1, in our framework relational content data is formalized in a (partially) shared data model and vocabulary based on RDF and RDFS, and stored in Sesame repositories. Standard RDF query languages (such as SeRQL or SPARQL) can be used to express queries using the least common denominator of the models and vocabularies of the used data sets. The results of these queries can be used as input for quantitative analysis, to retrieve the original articles for qualitative review, and visualized within the graphs that represent these articles.

This paper describes a use case, and provides a detailed experience report, of an intricate data analysis in a highly complex domain, with many non-trivial modeling decisions. It discusses the literature on a number of aspects where the representational and inferential requirements stretch the possibilities of the current standards. This case study serves both to underscore the use of Semantic Web technology to practically formalize a complex domain and to point out a number of issues on which there is still room for improvement.

In Sect. 2, we will briefly describe Relational Content Analysis and the NET method, and give a list of desiderata for a respresentation and inference mechanism for this domain. Section 3 is the main section of this paper, containing an overview of the existing options and work in progress on each point, and describes the modeling choices made to satisfy the requirements. Section 4 outlines the system that was created and the benefits that it has brought. +

2 Content Analysis as the Domain of Formalization

One of the goals of this paper is to find a suitable framework for formalizing the data produced by Relational Content Analysis (RCA). This section will outline what Content Analysis and RCA are, and then describe a particular method, the NET method, in more detail. Finally, a number of requirements for formalizing

this method will be listed, which will serve to guide the discussion in the next section.

2.1 (Relational) Content Analysis

Content Analysis is a social science method to analyze textual content by determining the occurrence of social scientific phenomena [1]. These phenomena are generally complex and subjective in nature, making the extraction a difficult task to automate (but see [5]). For this reason Content Analysis often uses human coders to read the text and directly indicate the presence of these phenomena.

Relational Content Analysis works by identifying smaller concepts, such as individual actors and issues, and coding the relations between these concepts as a graph [2]. The social scientific concepts are subsequently extracted as patterns or metrics defined over this graph. This two-step approach makes the data less dependent on the specific research question, creating a greater potential for sharing and reusing data.

Realizing this potential is difficult because we are dealing with data sets with heterogeneous data models and vocabularies. A formal representation that allows us to standardize both syntax and semantics of these data collections while remaining flexible enough to allow for different methods would be of great value in building the large data sets needed for statistically analyzing the complicated interaction between media and politics.

2.2 Relational Content Analysis Using the NET Method

The NET method [4] is the Relational Content Analysis method used in our case study. It has a fairly complex data model compared to other Relational Content Analysis methods [6]. Moreover, it includes a set of rules to make inferences about triples such as a form of transitivity. Furthermore, normal practice in NET is to annotate using very detailed concepts (such as 'Balkenende') and then aggregate to more general concepts (such as 'Politician') using a taxonomy.

In NET, sentences are coded as $<subject, predicate, object>$ triples. The *subject* and *object* are drawn from a predefined hierarchy of concepts. The predicate is complex, consisting of a *type*, and *quality*. The *type* indicates the kind of sentence code; possible types include 'causative', 'action', and 'affinitive.' *Quality* is a number that indicates the strength and direction of association between subject and object and ranges from -1 to 1.

Additionally, each triple can be augmented with two pieces of information. An *angle* can be specified for some statements, which captures the reason of a disagreement or action in sentences such as "Blair and Brown disagreed *about Iraq.* " Also, some sentences in a newspaper are quoted or paraphrased sources: "Blair stated that it was certain Saddam had WMD". In such sentences, the optional *quoted source* argument captures the source of the statement.

As an example, consider the newspaper excerpt in Fig. 2. The headline is coded as a reciprocal negative relation between the political blocks Left and Right. The first sentence of the lead is more complicated: The main message is that incumbent prime minister Balkenende (CDA / Christian Democrats) and

Hard confrontation Right and Left

The champions for the premiership, Labor leader Bos and Christian Democrat leader Balkenende, attacked each other over poverty and health care. Bos is needlessly scaring people, according to the prime minister. [..] Bos: "Good health care costs money, so we should invest more."

	Source	Subject	Relation	Object	Angle
1		Left	$\xleftrightarrow{-1 \text{ affinitive}}$	Right	
2		Bos	$\xleftrightarrow{-.7 \text{ affinitive}}$	Balkenende	Poverty
2		Bos	$\xleftrightarrow{-.7 \text{ affinitive}}$	Balkenende	Healthcare
3	Balkenende:	Bos	$\xrightarrow{-.7 \text{ acting}}$	Citizens	
4	Bos: Invest in Health		$\xrightarrow{+.5 \text{ causative}}$	Healthcare	
4		Bos	$\xrightarrow{+1 \text{ affinitive}}$	Invest in Health	

Source: De Telegraaf, 22 November 2006 (tr.auth). Reading: sentence 3 means that according to Balkenende, Bos is acting against the good of the citizens

Fig. 2. Example article with NET annotation

the challenger Wouter Bos (PvdA / Labor Party) are fighting, but it is also stated what they are fighting about: the issues Poverty and Health Care. In the next sentence, Balkenende states that Bos is scaring people, which is coded as Bos acting against the Dutch citizens with Balkenende as source. The final sentence expresses two relations: according to Bos, investing more money would be good for the Health Care, and Bos wants to invest money in Health Care, here coded as an affinity (issue position) relation between Bos and Health Care Investments.

Currently, NET-encoding of sentences is performed manually; research is performed to automate this but due to the complex nature of the information this has yet to lead to satisfying accuracy [7]. Given the cost of manual annotation, this difficulty in extraction only underscores the need to share and reuse existing data.

2.3 Requirements

This section will list a number of aspects of NET that we need to be able to capture in a formalization framework.

Representational Requirements. Relational Content Analysis methods use triples as the main primitive, but these triples are enriched in various ways.

R1: Background Knowledge A central feature of the NET method is that data is coded at a very detailed level, and aggregated to higher level theoretical concepts. This aggregation requires background knowledge, for example party memberships (Bos member-of Labor), political functions (Balkenende leads Cabinet), and is-a relations (FreeHealthCare is-a HealthIssue). This needs to be encoded and the concrete annotations need to be expressed in terms of this background knowledge. Moreover, although the taxonomies currently used in NET are purely hierarchical it would be useful if this could be relaxed. For example, Balkenende

is both a member of the CDA and the prime minister, and depending on the research question we want to use either fact for the aggregation.

R2: Statement types NET and other Relational Content Analysis methods use qualitatively different relations. For example, the statement "Bos and Balkenende attacked each other" is affinitive while "Good health care costs money" is causative. In Social Networks terminology these are called multiplex networks [8].

R3: Quantitative value In addition to different statement types, Relational Content Analysis often includes a quantitative indicator of the strength and direction of a relation. The statement ".. we should invest more" from the example is positive (+0.7) while the statement "Hard confrontation Left and Right" is strongly negative (-1). In Social Networks terms, graphs labeled with values are called signed and/or valued networks.

R4: Article Metadata To trace the evidence for an analysis and for time based analyses it is necessary to attach metadata to annotations, including publisher and publishing date, location in the newspaper, and a link to the original article.

R5: Extra Arguments Sometimes we need to code certain additional aspects of a relation. For example, in the sentence ".. attacked each other over poverty", we want to capture the topic of the disgreement as well as the fact that they disagree.

R6: Quoted Sources The example sentence "Bos: 'good health care costs money, so we should invest more" contains a positive causal relation between Investing and Health Care, but this relation is not directly stated by the newspaper but rather by a quoted source. In order to analyze such sentences correctly, it is necessary that the contained triples are accessible for analysis, but they should be kept separate from the main graph.

Usage Requirements. The requirements above all specify what kind of information we need to be able to represent. Next to these requirements there are also a number of things we need to be able to do, mainly while analyzing the annotated media material. These 'usage' requirements are outlined below.

R7: Shareability One of the purposes for formalizing NET is to make it easier to share and combine data sets. Therefore, it has to be possible to combine data sets that differ both in exact data structure and in the used vocabulary.

R8: Time-bound roles In Content Analysis, the social and political roles played by actors are generally considered background knowledge and remain static during a project. However, social roles are dynamic and especially for longitudonal analyses we need to be able to represent the temporal validity of political roles

R9: Disjoint Categorization Often, we want to aggregate the nodes in our media data to higher-level categories. These categorizations generally have to be disjoint and exhaustive. This is necessary to avoid counting one instance twice and is also assumed by many statistical analyses. Therefore, we want to be able to express disjoint categorizations and to check or prove that a categorization scheme is disjoint and exhaustive given the structure of the background knowledge.

R10: Extraction As stated above, it is often useful to categorize the nodes in the relational network into higher level categories. Therefore, it is necessary to have a formalization that allows extracting the network data using such categorizations.

3 Formalizing the NET Method

As described above, the NET method is a Relational Content Analysis (RCA) method with a fairly complex data model and usages. The relational nature of NET and the need to combine and share data sets with different structure and vocabulary make Semantic Web technologies a logical choice for formalizing this domain. This section will describe the modeling choices we made to meet the requirements listed in the section above.

3.1 R1-2, R7: Low Hanging Fruit

Due to their relational nature, Semantic Web technologies seem a natural match for the formalization of RCA. This was confirmed by a number of requirements that were fulfilled easily and elegantly.

Background Knowledge (R1) Background knowledge can be expressed elegantly using RDF(S), using either *rdfs:SubClassOf* statements or custom vocabulary. RDF(S) places no restriction on the types and amount of relations between concepts, allowing for multiple inheritance and different relation types.

Statement types (R2) In RDF, the predicative part of triples consists of an RDF resource that can be described in the same way as other resources. This means that it is natural to express multiplex networks in RDF.

Shareability (R7) RDF(S) does not solve the conceptual and substantive problems of combining heterogeneous data sets. It does, however, remove a number of technical difficulties. Globally unique names using URI's minimize vocabulary clashes while the subclass and subproperty mechanisms facilitate mapping specific vocabulary onto more general terms.

3.2 R3-6: Enriching Triples with Extra Information

Requirements R3-6 boil down to a single wish: enriching triples by adding extra information. This is difficult, as RDF is meant to describe resources, not triples: triples do not have URIs and hence cannot be part of triples. We are not the first to signal this difficulty: [9] cites the need for enriching triples to describe event data, and a number of authors want to use RDF for describing RDF documents, for example for reasoning about provenance and trust [10].

RDF(S) allows some form of adding information to existing triples. Trivially, we can replace each of the nodes in a triple by a node carrying more information and point to the original node. In RDFS, it is possible to do so transparently by making the new node a subclass or subproperty of the original node. Additionally, the RDFS specification includes a reification mechanism [11]. Essentially, an anonymous instance is made to represent the statement, and standardized vocabulary is used to define the subject, object, and predicate of the relation. The anonymous instance, being a first class citizen, can then be used in further statements. According to the definition, a reified statement does not necessarily imply the original statement: it is describing a hypothetical event.

Another solution is using the n-ary relation design patterns described in [12]. This is similar to reification in that a new node is created that represents the relationship, but the reification vocabulary is eschewed since "in n-ary relations [..] additional arguments in the relation do not usually characterize the statement but rather provide additional information about the relation instance itself" [12]. This has the same disadvantage as reification (the original triple semantics are lost) but additionally it has no formal meaning or standardized vocabulary.

To overcome these problems, a number of authors have suggested extending the notion of a triple to include a fourth place, often seen as a context marker [9,10,13,14,15]. For example, [14] propose a context mechanism that explicitly assumes the context marker to indicate provenance and they include a complicated system of lifting and aggregating mechanisms to combine RDF documents from different sources. On the other extreme, [9,13] support replacing triples by quadruples without restricting the interpretation of such triples.

A proposal that seems to be gaining ground is Named Graphs [10]. This proposal also adds a fourth places to the triple and defines the semantics of this added element but does not prescribe the interpretation in the way [14] does. Named Graphs semantics allow for nested graphs and they propose a predicate for indicating nesting. The main disadvantage of this method is that it is not standardized, leaving tool support and declarative semantics to be desired. Also, as the intended meaning of the context is the containing graph, Named Graphs add extra information to the whole statement rather than to the predicate much like reification does.

The proposals for adding information to triples in the literature are diverse. Part of the reason for this diversity is that the problem they are trying to solve is diverse. We think that there are two main factors on which the proposed solutions diverge: the *meaning* of the extra information with respect to the original triple; and the *opacity* of the enrichment. In terms of meaning, we distinguish four possible relations of the new information x to the existing triple Rab:

- $\boldsymbol{R^x ab}$: Adding information about the predicate of the triple;
- $\boldsymbol{Ra^x b}$: Adding information about the subject or object of the triple;
- $\boldsymbol{(Rab)^x}$: Adding information about the whole triple; and
- $\boldsymbol{Rabx}$: Adding an extra argument to the triple on equal footing with the subject and object.

In terms of opacity, we distinguish between transparent and opaque additions:

- **Transparant** additions preserve the original meaning of the triple in the graph, meaning that applications that do not interpret the richer relation can still see the original relation; while
- **Opaque** additions remove the original triple from the graph, meaning that it will not be visible to an application that does not (or cannot) interpret the enrichment technique.

Depending on the modeling requirements, we want to add certain information to a triple in a certain way. For example, a quoted source in a newspaper

should be an opaque statement about the whole triple, while quality should be a transparant addition to the predicate. Thus, rather than looking for a single 'correct' solution we think that multiple options are needed to express these differences in enrichment. Table 1 categorizes the discussed proposals in these terms and serves as the basis for making the appropriate modeling choices. The proposal by [14] is left out of this table because its purpose is strictly describing graphs rather than enriching triples

Table 1. Suitability of discussed mechanisms for expressing different triple enrichments

	Transparent				Opaque			
	Enriching an argument Ra^xb	Enriching the predicate R^xab	Enriching the triple $(Rab)^x$	Extra argument $Rabx$	Enriching an argument Ra^xb	Enriching the predicate R^xab	Enriching the triple $(Rab)^x$	Extra argument $Rabx$
RDF					$\pm^1$	+		
RDFS	+	$\pm^1$			$\pm^1$	+		
N-ary							$\pm^2$	+
Reification							+	$\pm^2$
Quadruples			$\pm^3$	$\pm^3$				
Named Graphs			+				+	

[1]Adding a dicrete categorization is possible, but adding quantitative information is very difficult. [2]N-ary patterns are explicitly intended to express an extra argument to a statement, while reification is intended to express information about a statement, making other use of these solutions difficult to interpret. [3]Since there is no specified interpretation of the extra argument it is not possible to distinguish between these two cases.

We will now reconsider the requirements from the previous section in terms of Table Table 1. As listed in the previous sections, the basic unit of information is a triple representing a media relation (eg. Bos dislikes Balkenende). To this triple we add information to *quantify* (R3) the predicate, add *extra arguments* (R5) to the relation, *specify the source of a quoted statement* (R6), and link the media statement to *metadata* (R4) such as publisher and publishing date. As stated above, quoted sources should be opaque as the quoted statement is not directly asserted by the newspaper. The other additions should all be transparent: the original triple is a valid part of the graph with or without the extra information. The *quantification* is an enrichment of the predicate, but very difficult to represent using subproperties because of the quantitative and unrestricted nature of the information. The *extra arguments* and *quoted source* both add extra arguments that are subordinate to the main triple, falling somewhere between the intended meaning of reification (statements about triples) and n-ary relations (multiple arguments of equal weights). The *metadata* is adding information to the whole triple, and fits in the use case of reification and named graphs.

Surveying the table above, there is no perfect method for adding information to triples. Named graphs have the desired transparency but offer no solution for distinguishing between extra arguments and metadata. Quadruples allow extra arguments in a natural way but this comes at the expense of flexibility and semantic clarity. Within the existing standards, reification covers adding metadata

and a case could be made for using reification to represent additional arguments. N-ary relations are better suited for the additional arguments but suffer from the lack of a standard vocabulary. It is possible to mix and match mechanisms, but this comes at the expense of increasing complexity and if multiple non-standard mechanisms are mixed it will be difficult for third parties to understand what we mean.

Solution. For the current application, we decided to stick to one representation for all enrichments. Since tool support for the proposed extensions is still limited, and the intended meaning of our enrichment is closer to meta-statements than to adding arguments, we decided to use RDFS reification.

3.3 R8: Dynamic Roles

Social Roles, such as being a party member or fulfilling a political function, are a complex topic that has received extensive attention in the literature [16,17,18,19, 20]. As described by [16], two defining characteristics of roles are that they are anti-rigid[1], and dynamic[2]. In this definition, background knowledge such as party membership and political office can be classified as knowledge on role memberships of actors.

[17] surveys a large body of literature and notes that there are three main approaches to representing roles. The first approach is calling the places in a predicate roles, i.e. in a predicate $memberOf(member, group)$ the roles *member* and *group* are implied. This corresponds to creating a simple RDF relation between the member and the group. Using this mechanism, it is impossible to represent temporal constraints on roles, and such statements should be considered snapshots of a dynamic relation rather than descriptions.

The second approach is to make the role a subtype of the natural type corresponding to it. This means that playing the role of being a PvdA member means creating a subclass of politician, the PvdAMemberPolitican. As described in [17], this leads to a number of complications and does not solve the representation problem inherent in the first approach.

The third approach is creating an *adjunct instance* representing the relation, which is an instance of the role type but unique for each instantiation. Since this promotes the role membership to first class citizen, it allows for further specification such as temporal aspects. In RDF, this can be described as a (blank) node representing the membership, with relations to the two role players and the role type, which is also the approach taken by in [21]. Interestingly, if the RDF reification vocabulary is used to denote the integral aspects of the role, this is equivalent to reifying a simple statement expressing the relation directly.

Solution. In the terminology introduced in section 3.2, we want the enrichment of the original $memberOf$ predicate to be opaque since the roles are invalid outside their (temporal) context. This makes creating *adjunct instances* by using reification a natural choice for representing social roles.

[1] The role players do not depend on their playing the role for their existence.
[2] Roles change over time and there is no 1-on-1 relation between roles and players.

3.4 R9: Disjoint Categorization

As described above, adding background knowledge and using this knowledge to link 'data level concepts' to 'theory level concepts' can be done elegantly using RDF. A frequent use case in Content Analysis is to define a set of categories on the media data, for example statements with an opposition politician as subject, with a coalition politician as subject, and statements with a societal actor as subject. Counts of such statements per period are then used either in statistical analysis or presented in a table. Both uses require the categories to be disjoint and exhaustive with respect to the higher category, in this case 'actor statements'. In other words, the higher category should be *partitioned* by the proposed categories.

Using model checking (e.g. SPARQL queries), it is trivial to *check* whether a categorization, expressed as a set of requirements, is a partitioning given a concrete data set. By pair-wisely conjoining the requirements disjointness can be checked, and by negating the whole conjunction exhaustiveness can be checked.

In some cases, such as presenting data real-time on a web page, we would like to be able to *prove* that such a categorization will always be a partitioning. In RDF this is impossible due to the fact that cardinality, disjointness, and negation cannot be asserted, so it is impossible to express the constraint that a politician is a member of exactly one party or that societal actors are all non-political actors. In OWL these restrictions can be expressed, and proving disjointness boils down to proving that each pairwise conjunction of the categories is unsatisfiable. Exhaustiveness can be shown by proving that the higher category implies membership of one of the lower categories. More formally, proving that the categories $\{A_1 \dots A_n\}$ partition B in the ontology $\mathcal{O}$ means proving $\mathcal{O} \models A_i \sqcap A_j = \bot$ for all $i \neq j$, $i, j \leq n$ and $\mathcal{O} \models B \sqsubseteq A_1 \sqcup \dots \sqcup A_n$.

Solution. For the current application, we decided to stay within RDF and only use query-based model checking of the disjointness.

3.5 R10: Categorizing and Extracting Data

As described above, it is useful to define categorization schemes and aggregate the media data to a higher level using such schemes. A scheme will generally consist of a high level category and all its instances and parts and members of these instances. However, as described in Sect. 3.3, these part-of and member-of relations will often be dynamic and represented using adjunct instances. Therefore, we need to check whether a role is actually valid at the publishing date of the article. Moreover, as described in Sect. 3.4, it is often necessary to include negations in the definition of categorization to prevent actors with multiple roles from being counted twice.

This leads to a complex definition for these categories. Since they will often include negation, they cannot be represented in RDF(S). It would be possible to represent them in OWL, but this requires complex concrete domain reasoning for the date comparisons. Practically, it is possible to conduct such categorizations using closed-world model checking in RDF, for example using a SeRQL query.

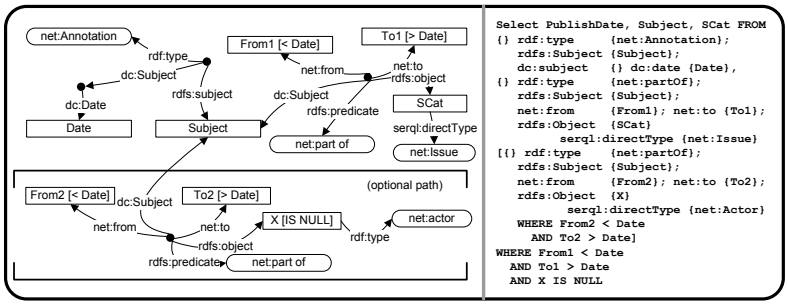

Fig. 3. The (partial) extraction query, represented visually and in SeRQL

This results in a query such as shown in Fig. 3, where the Subject of a statement is categorized as an issue but only if it is not categorizable as an Actor. To create the real query, this has to be duplicated for subject and object and a UNION query has to be created joining all category definitions.

Solution. For the applications described below we used SeRQL queries to extract data, using query rewriting to hide some of the query complexity from the user. If we describe the categorizations in OWL it should be possible to automatically rewrite the OWL definitions into RDF queries (assuming a closed world), or cache the categorization results from the DL reasoner.

4 Implementation

The sections above outlined the challenges encountered while modeling the Relational Content Analysis domain and the possible solution for these challenges. This section will briefly describe the actual systems that were built around the RDF representation, especially the annotation tool, the browser/visualizer and the extractor.

4.1 Data Model and Ontology

This section will describe the data model that resulted from the choices described in Sect. 3. Fig. 4 visualizes this model. The main element of the data model, the original triple of the relational method, is now a reified triple, making the annotation (a subclass of `rdf:Statement`) the central element. Annotations have a subject, predicate and object as required for reification, and also have the quantitative value 'connection' and an angle. On the left hand side, annotations are connected to textual units (sentences) from an article using the `dc:subject`, and metadata about this article are recorded. Additionally, the coder, the creator of the annotation, is recorded.

On the right hand side the subject, object, and predicate are all drawn from the ontology, having `net:entity` as its base. This ontology contains an IS-A hierarchy of (political) actors and issues together with role information such as party

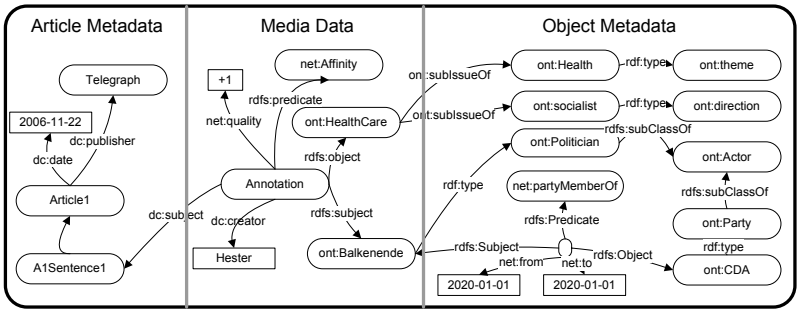

Fig. 4. The data model used

membership. As described in Sect. 3.3, these roles are made dynamic by reifying the role membership statement, creating an adjunct instance, and adding from and to dates. The ontology is a formalization of an existing taxonomy, containing 478 actors in 32 (nested) categories and an issue hierarchy of 103 issues.

4.2 The iNET Annotation Tool

A new version of the existing iNET tool was created in Java Eclipse for annotating newspaper articles in this framework. As can be seen in Fig. 5, iNET shows

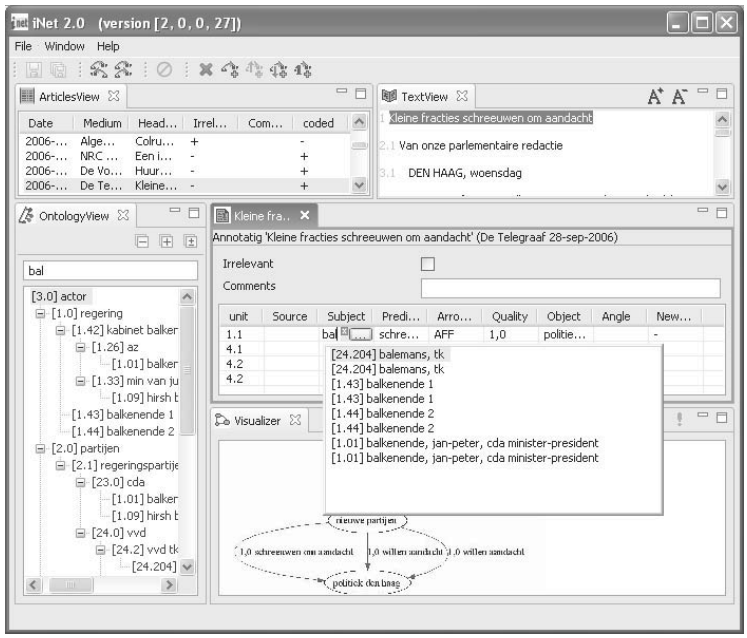

Fig. 5. iNET: RDF-based annotation with Autocomplete and Visualization

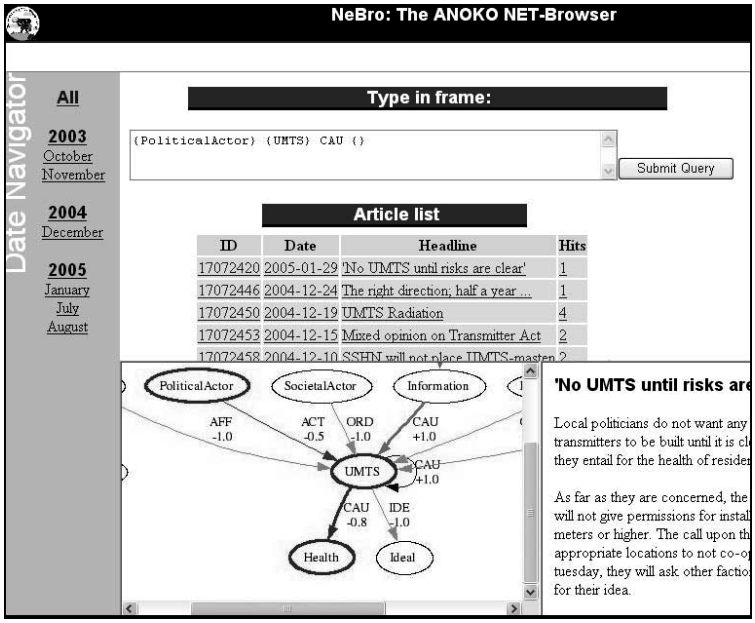

Fig. 6. NeBro: Browsing, Querying, and Visualizing the RDF repository

the content of the current article and the ontology, and assists annotators by offering autocomplete functionality and by allowing search through the ontology. A coder logs into a (relational) article database, which gives him or her a list of jobs, consisting of the to-be-annotated articles and the annotation scheme and a pointer to the ontology in the Sesame repository. After coding an article, results are immediately posted to that repository. To help coders check their annotations, a visual representation of their coding is also presented. Using this tool, a group of 14 coders annotated over 13,000 articles and tv news broadcasts, resulting in 30,000 triples.

4.3 The NeBro Browser / Visualizer

To browse through these results, a web application was created that allows a user to browse through articles and view a visualization of the annotation. As shown in Fig. 6, a user is able to enter queries to look for specific relations. Since the internal representation became quite complex, these queries can be posted in terms of the original NET relations and are translated to SeRQL queries on the Sesame Repository. In the visualization of the retrieved article, the relations matching the query are highlighted.

5 Conclusions

In this paper we reported our experiences in designing a formal representation for Relational Content Analysis, a method used in Communication Science to

conduct quantitative media analysis. This domain, and particularly the NET methodology used in a recent use-case on the Dutch election campaign in 2006, highlighted a number of different requirements on representation and querying which could only partly be implemented seamlessly using current Semantic Web technology.

The main contribution of this paper is threefold:

1. The paper describes a real world, large size case-study where Semantic Web technology was used for representing and querying highly complex data on media coverage on the Dutch election campaign 2006, and thus exemplifies a state of matureness of the technology. Based on a detailed requirement analysis we designed workable and flexible data models to code the Relational Content Analysis data. On some aspects we identified conceptual problems that require more general solutions which should lead to extensions of existing standards in the future.
2. The paper identifies a number of such requirements to a formal representation framework, which are domain and application specific at first instance. Nevertheless, we expect the majority of these requirements to be recurrent in many other practical applications, and although we do not claim exhaustiveness the collection of items in Sect. 2.3 will be indicative for problems faced in similar applications.
3. For each of the requirements, this paper provides an overview of the state of the art in practical Semantic Web research. We surveyed the literature on extending RDF triples with additional information, and presented a categorization of existing and proposed mechanisms in terms of the meaning of the extra information and the transparency of the original triple. We hope that such a study (from an application perspective) can pinpoint relevant open research questions for the Semantic Web community.

The general conclusion is that Semantic Web technology offers a useful set of standards and tools for formalizing this background knowledge and storing and inferencing with the combination of background knowledge, metadata about the annotated articles, and the annotations themselves. Having all this data within one representation enabled us to develop the set of tools presented in this paper for efficiently annotating, visualizing, querying, and extracting from the data set.

References

1. Krippendorff, K.: Content Analysis: An Introduction to Its Methodology (second edition). Sage Publications (2004)
2. Carley, K.: Network text analysis: The network position of concepts. In Roberts, C., ed.: Text Analysis for the Social Sciences. Lawerence Erlbaum Associates, Mahwah, NJ (1997) 79–100
3. Kleinnijenhuis, J., Scholten, O., van Atteveldt, W., van Hoof, A., Krouwel, A., Oegema, D., de Ridder, J.A., Ruigrok, N., Takens, J.: Nederland vijfstromenland: De rol van media en stemwijzers bij de verkiezingen van 2006. Bert Bakker, Amsterdam (2006)

4. Van Cuilenburg, J.J., Kleinnijenhuis, J., De Ridder, J.A.: Towards a graph theory of journalistic texts. European Journal of Communication **1** (1986) 65–96
5. Wiebe, J.M., Wilson, T., Bruce, R.F., Bell, M., Martin, M.: Learning subjective language. Computational Linguistics **30(3)** (2004) 277–308
6. Van Atteveldt, W., Kleinnijenhuis, J., Carley, K.: Rcadf: Towards a relational content analysis standard. In: Presentated at the International Communication Association (ICA), Dresden (2006)
7. Van Atteveldt, W., Oegema, D., van Zijl, E., Vermeulen, I., Kleinnijenhuis, J.: Extraction of semantic information: New models and old thesauri. In: Proceedings of the RC33 Conference on Social Science Methodology, Amsterdam (2004)
8. Wasserman, S., Faust, K.: Social Network Analysis. CUP, Cambridge (1994)
9. MacGregor, R., Ko, I.Y.: Representing contextualized data using semantic web tools. In: Practical and Scalable Semantic Web Systems (workshop at second ISWC). (2003)
10. Carroll, J., Bizer, C., Hayes, P., Stickler, P.: Named graphs, provenance and trust. In: Proceedings of the Fourteenth International World Wide Web Conference (WWW2005), Chiba, Japan. Volume 14. (2005) 613–622
11. Brickley, D., Guha, R.: Rdf vocabulary description language 1.0: Rdf schema. W3C Recommendation (http://www.w3.org/TR/rdf-schema/) (2004)
12. Noy, N., Rector, A.: Defining n-ary relations on the semantic web. Working Draft for the W3C Semantic Web best practices group (2005)
13. Dumbill, E.: Tracking provenance of rdf data. Technical report, ISO/IEC (2003)
14. Guha, R., McCool, R., Fikes, R.: Contexts for the semantic web. In: Proceedings of the Third International Conference on the Semantic Web (ISWC'04). (2004)
15. Sintek, M., Decker, S.: Triple - a query, inference, and transformation language for the semantic web. In: Proceedings of ISWC02. (2002)
16. Masolo, C., Vieu, L., Bottazzi, E., Catenacci, C., Ferrario, R., Gangemi, A., Guarino, N.: Social roles and their descriptions. In Dubois, D., Welty, C., Williams, M., eds.: Proceedings of the Ninth International Conference on the Principles of Knowledge Representation and Reasoning (KR2004), Whistler, Canada (2004) 267–277
17. Steimann, F.: On the representation of roles in object-oriented and conceptual modelling. Data and Knowledge Engineering **35** (2000) 83–106
18. Sowa, J.: Using a lexicon of canonical graphs in a semantic interpreter. In Evens, M., ed.: Relational models of the lexicon. Cambridge University Press, Cambridge UK (1988)
19. Sowa, J.: Knowledge Representation: Logical, Philosophical, and Computational Foundations. Brooks/Cole, Pacific Grove, CA (2000)
20. Guarino, N.: Concepts, attributes and arbitrary relations: Some linguistic and ontological criteria for structuring knowledge bases. Data and Knowledge Engineering **8** (1992) 249–261
21. Mika, P., Gangemi, A.: Descriptions of Social Relations. In: Proceedings of the 1st Workshop on Friend of a Friend, Social Networking and the (Semantic) Web. (2004)