ONTOLOGY-BASED QUERY REFINEMENT FOR MULTIMEDIA META OBJECTS

Sonja Zillner and Werner Winiwarter

Department of Computer Science and Business Informatics
University of Vienna, Austria
{sonja.zillner,werner.winiwarter}@univie.ac.at

Abstract: *To enable efficient access to multimedia content, the media data has to be augmented by semantic metadata and functionality. The semantic representation has to be integrated with domain ontologies to fully exploit domain-specific knowledge. This knowledge can be used for refining ambiguous user queries by closing the conceptual gap between the user and the information to be retrieved. In our previous research, we have introduced Enhanced Multimedia Meta Objects (EMMOs) as a new approach for semantic multimedia meta modeling, as well as the query algebra EMMA, which is adequate and complete with regard to the EMMO model. This paper focuses on the refinement of EMMA queries by incorporating ontological knowledge.*

1 Introduction

Although more and more multimedia content becomes available, searching and retrieving multimedia resources is still an open and critical issue. Text-based keyword search techniques alone are not sufficient for retrieving multimedia content. One way to improve the search for multimedia content is to enhance multimedia resources by semantic metamodels and to integrate domain ontologies.

To answer these challenges, we have developed *Enhanced Multimedia Meta Objects (EMMOs)* [14], a novel approach for semantic multimedia content modeling. EMMOs were created within the EU-project CULTOS to model InterTextualThreads (ITTs), i.e. complex knowledge structures used by the researchers in intertextual studies to share and communicate their knowledge about relationships between cultural artefacts. An EMMO establishes a self-contained piece of multimedia content that indivisibly combines three of the content's aspects. The *media aspect* describes that an EMMO assembles the media objects of which the multimedia content consists, the *semantic aspect* specifies that an EMMO further encapsulates associations between its media objects, and the *functional aspect* defines operations on the content and the semantic description of an EMMO that can be invoked by applications. Moreover, EMMOs are *tradeable*, i.e. EMMOs can be packaged and exchanged in their entirety including media, semantics, and functionality. Finally, EMMOs constitute *versionable* units that can be concurrently modified and authored within a distributed collaborative environment.

For the efficient retrieval of and access to the multimedia content within EMMOs, we have developed the query algebra *EMMA*, which is adequate and complete with regard to the EMMO model, i.e. it provides the required operators to enable the access to the complete information stored within the EMMO model. By defining simple and orthogonal operators that can be combined to build complex queries, EMMA enables query optimization. However, the retrieval performance can be further improved significantly by integrating the EMMO model with a domain ontology.

The main contribution of this paper is to show how the ontological knowledge can be used for query refinement within the EMMO model. By describing semantic associations between its media objects, an EMMO establishes a graph-like knowledge structure with associations and nodes being labeled by concepts of the ontology. Within EMMA, we have defined navigational operators to provide means to traverse the ontology-labeled associations within an EMMO's graph structure. Regarding the integration of ontology knowledge, we focus our efforts on *relational concepts*, i.e. concepts used for labeling associations, because ontological knowledge about these concepts is essential for enriching the expressive power of navigational operators, and thus for query refinement. We will show that the incorporation of ontological knowledge can be achieved either explicitly by extending the graph structure of an EMMO with additional associations or implicitly by extending the navigational operators in EMMA.

The remainder of the paper is organized as follows. In Sect. 2, we discuss related approaches and standards. Section 3 gives a brief overview of the EMMO model, and Sect. 4 presents EMMA's operators for navigating an EMMO's graph structure. In Sect. 5, we define the requirements for an ontology of relational concepts, and in Sect. 6, we describe how the ontology knowledge can be integrated into the EMMO model for query refinement. Section 7 concludes this paper with an outlook on future work.

2 Related Work

By incorporating the media, the semantic, and the functional aspect, as well as versioning support in a homogenous way, the EMMO model constitutes a unique approach to multimedia meta modeling. None of the standards for *multimedia document models*, such as SMIL [1] or SVG [7], and none of the *standards for semantic media description*, such as RDF [13] or Topic Maps [11], addresses all these aspects. Thus, none of the query languages for those standards can fulfil all requirements with respect to the expressiveness of a query language for EMMOs. However, valuable aspects of their design have been incorporated into the design of the query algebra EMMA.

Our main concern is the enhancement of the expressive power of EMMA's navigational operators by integrating ontology knowledge. Therefore, we analyzed query languages for standards for semantic media description, such as RDF or Topic Maps, regarding whether they allow for navigation of the graph structure and whether the navigational access can be refined by integrating ontology knowledge.

Existing query languages for *RDF*, such as RAL [9] or RQL [12], provide means to navigate the RDF graph structure and enable the integration of a very simple ontology structure described by RDF Schema [4]. However, they cannot deal with more elaborate ontology constructs, such as the transitivity or symmetry of relationships. The situation for query languages for *Topic Maps* is quite similar as for RDF. Approaches such as Tolog [10] or TMPath [3] provide means for graph navigation, but only Tolog addresses ontology integration by offering some very limited features, e.g. querying concept-subconcept relationships.

Although there exist quite a few approaches for ontology-based browsing of text collections, such as QuizRDF [5], Spectacle [8], or SESQ [16], none of those approaches is suitable for retrieving multimedia data or allows to refine user queries by drawing inferences from ontological knowledge. However, those approaches offer useful insights into the use of ontology knowledge for visualization and navigation in browsing interfaces, which we will bear in mind for future work on designing the user interface of an ontology-based search engine.

3 The EMMO Model

As mentioned before, an EMMO is a self-contained unit of multimedia content that unifies three aspects, i.e. the media, the semantic, and the functional aspect, and provides versioning support. Figure 1 shows the EMMO "Dracula", which we will use as a running example throughout this paper. The formal basis of the EMMO model are *entities*, which occur in four different specializations:

- *ontology objects* represent concepts of an ontology,
- *logical media parts* represent media objects or parts of media objects, e.g. video scenes or book chapters,
- *associations* model binary relationships,
- *EMMOs* aggregate semantically related entities.

Each entity is globally and uniquely identified by its OID and described by an arbitrary number of attribute-value pairs with the attribute name being a concept of the ontology; e.g. the ontology object *Director* describes an attribute of the logical media part "Dracula", which has the string value "F. Coppola". For enabling *versioning support*, each entity can refer to an arbitrary number of preceding and succeeding versions.

To model an EMMO's *media aspect*, each logical media part is characterized by a set of connectors with each connector pointing to a physical representation of the media object, e.g. the EMMO "Dracula" contains the text documents "Frankenstein.doc", "DraculaSchool.pdf", and "DraculaBite.pdf", as well as the two videos "Vampir.avi" and "Dracula.mpeg".

By specifying a set of operations, EMMOs address the *functional aspect*, e.g. EMMO "Dracula" provides a rendering operation, which returns a presentation of the EMMO's content in

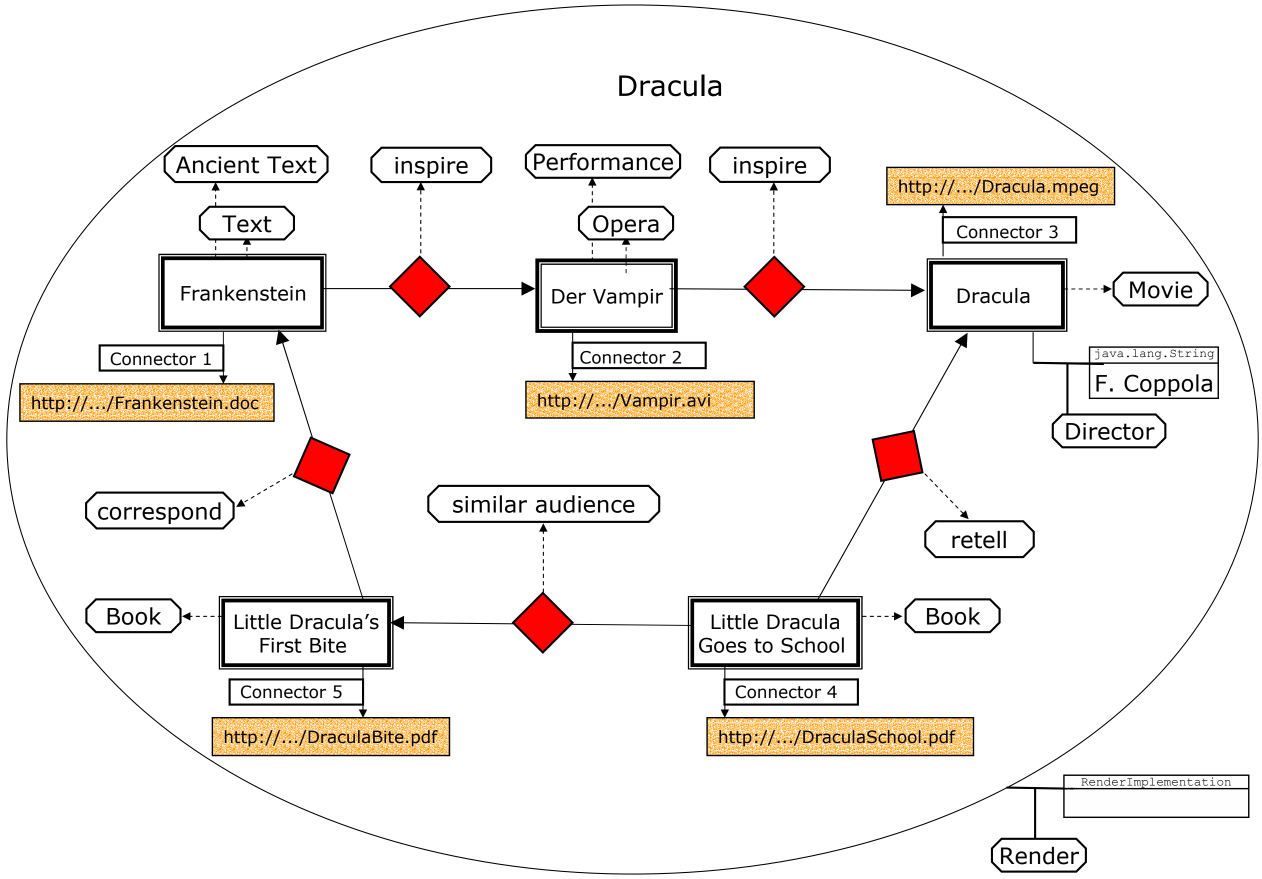

Figure 1: EMMO "Dracula" ($e_{dracula}$)

different formats, such as HTML or PDF.

For the representation of an EMMO's *semantic aspect*, each EMMO is described by a *nodes* set, specifying all the entities contained within an EMMO. Semantic relationships between entities are modeled as directed associations by specifying a source and target entity for which the relationship holds. Therefore, EMMOs constitute a graph-like knowledge structure with links and edges being labeled by ontology objects representing concepts of the domain ontology. The media objects contained within EMMO "Dracula" are digital manifestations of the ancient text "Frankenstein" written by Mary Shelley, the German opera "Der Vampir", the movie "Dracula" directed by Francis Coppola, and the two children books "Little Dracula Goes to School" and "Little Dracula's First Bite". The types of the media objects are defined through a reference to concepts of the domain ontology, e.g. *Ancient Text* or *Performance*. By labeling the associations with relational concepts from the ontology, we can indicate that the historical text "Frankenstein" *inspired* the German opera "Der Vampir", which again *inspired* the movie "Dracula". In addition, we can express that both books "Little Dracula Goes to School" and "Little Dracula's First Bite" address a *similar audience*, "Little Dracula Goes to School" *retells* the "Dracula" movie, and "Little Dracula's First Bite" *corresponds* to the "Frankenstein" text.

4 Navigational Operators in EMMA

The query algebra EMMA defines five general classes of query operators. The *extraction operators* provide means to access all the attributes of the entities within the EMMO model, the *selection predicates* allow the selection of only those entities satisfying a specific characteristic, the *construction operators* enable the modification, combination, and creation of new EMMOs, and the *join operator* relates several entities or EMMOs with a join condition. Finally, the *navigational operators* enable the navigation along an EMMO's semantic graph structure, and thus provide the basis for ontology-based query refinement. Therefore, in the following, we will focus on a more detailed description of navigational operators.

Navigation through an EMMO's graph structure is controlled by a *navigation path* defined as a set of *sequences* of ontology objects. A mapping for each ontology object in a sequence to the corresponding association within an EMMO defines the traversal of the graph structure.

We have defined *regular path expressions* over ontology objects for describing the syntax of a navigation path. Navigational operators take a regular path expression as input and specify how those syntactic expressions are applied to navigate the graph structure. For example, for a given EMMO, start entity, and regular path expression, the navigational operator *JumpRight* returns the set of all entities that can be reached by traversing the navigation path in the right direction, i.e. by following associations from source to target entities. Applying the operator *JumpRight* to the EMMO "Dracula" ($e_{dracula}$) , the starting entity "Frankenstein" (l_{frank}), and the primitive regular path expression consisting of one single ontology object *inspire* ($o_{inspire}$) yields the logical media part representing the video of the opera "Der Vampir" (l_{vampir}):

$$JumpRight(e_{dracula}, l_{frank}, o_{inspire}) = \{l_{vampir}\}.$$

In addition to *one single ontology object*, there exist two other primitive regular path expressions:

- the *empty expression* "ε" refers to the empty entity and is interpreted by a navigational operator as absence of movement,
- the *wildcard expression* "_" refers to any arbitrary ontology objects and is interpreted by a navigational operator as following any arbitrary association regardless of the labeling.

There exist two ways of combining regular path expressions:

- regular path expressions can be *concatenated* to create a longer navigation path,
- the *union operator* "|" allows to treat several regular path expressions as alternative branches.

Finally, we have defined four unary operators for the modification of regular path expressions:

- adding the operator "?" to a regular path expression specifies its *optionality*,
- adding the operator "+" to a regular path expression specifies an *iteration* of path expressions, which is interpreted as navigation along the same regular path expression

any number of times but at least once,

- adding the *Kleene Star operator* "$*$" to a regular path expression specifies an iteration of path expressions, which is interpreted as navigation along the same regular path expression any number of times (including the absence of movement),
- adding the operator "$-$" to a regular path expression expresses the *inversion* of the regular path expression, i.e. the change of direction of navigation.

Table 1 shows examples of applying the *JumpRight* operator with different types of regular path expressions to the EMMO "Dracula"($e_{dracula}$). The symbols l_{frank}, l_{vampir}, $l_{dracula}$, l_{bite}, and l_{school} designate the logical media parts "Frankenstein", "Der Vampir", "Dracula", "Little Dracula's FirstBite", and "Little Dracula goes to School"; $o_{inspire}$, $o_{similar}$, $o_{correspond}$, and o_{retell} the ontology objects *inspire*, *similar audience*, *correspond*, and *retell*.

Table 1: Examples of Applying Regular Path Expressions

Basic Pattern	Example Query	Query Result
o_i	$JumpRight(e_{dracula}, l_{frank}, o_{inspire})$	$\{l_{vampir}\}$
ε	$JumpRight(e_{dracula}, l_{frank}, \varepsilon)$	$\{l_{frank}\}$
$\_$	$JumpRight(e_{dracula}, l_{school}, \_)$	$\{l_{bite}, l_{dracula}\}$
$o_i o_j$	$JumpRight(e_{dracula}, l_{school}, o_{similar} o_{correspond})$	$\{l_{frank}\}$
$o_i \mid o_j$	$JumpRight(e_{dracula}, l_{school}, o_{similar} \mid o_{retell})$	$\{l_{bite}, l_{dracula}\}$
$o_i?$	$JumpRight(e_{dracula}, l_{frank}, o_{similar}? o_{inspire})$	$\{l_{vampir}\}$
o_i+	$JumpRight(e_{dracula}, l_{frank}, o_{inspire}+)$	$\{l_{vampir}, l_{dracula}\}$
o_i*	$JumpRight(e_{dracula}, l_{frank}, o_{inspire}*)$	$\{l_{frank}, l_{vampir}, l_{dracula}\}$
o_i-	$JumpRight(e_{dracula}, l_{dracula}, o_{retell}-)$	$\{l_{school}\}$

The inversion of a regular path expressions, i.e. the traversal along the opposite direction, can also be expressed with the navigational operator *JumpLeft*, e.g.:

$$JumpLeft(e_{dracula}, l_{frank}, o_{correspond}) = JumpRight(e_{popular}, l_{frank}, o_{correspond}-) = \{l_{bite}\}.$$

5 Requirements for an Ontology of Relational Concepts

By providing a shared and common understanding of a domain that can be communicated between people and applications systems, ontologies facilitate the sharing and reuse of knowledge [6]. An ontology suitable for the integration into the EMMO model has to distinguish between *object concepts* and *relational concepts*. Object concepts are used for labeling the nodes, relational concepts for labeling the associations within an EMMO's graph structure. For example, the Ontology of Intertextuality [2], which was established within the CULTOS project, defines object concepts for describing media objects, e.g. the concepts *Text* or *Performance*, and relational concepts for describing relationships holding between media objects; e.g. the relational concept *inspire* can be used for describing the fact that a historical source text *inspires* a particular performance.

In the following, we will focus on relational concepts, because those concepts are essential for enhancing the expressive power of navigational operators, and thus provide the basis for ontology-driven query refinement. Within an ontology structure, relational concepts can be characterized by additional features. We identified the following four important characteristics, which are relevant for query refinement. Relational concepts can be defined as:

- *subconcepts* of other relational concepts, thus establishing a transitive subconcept hierarchy,
- *transitive concepts*, i.e. concepts for which an iteration of the corresponding path expression can be defined without changing the semantics of the concepts,
- *symmetric concepts*, i.e. concepts for which all associations can be traversed in both directions, i.e. source and target entities can be exchanged without changing the semantics of the concept,
- *inverse concepts* to other concepts, i.e. if the label of an association is changed to the inverse concept, then source and target entity have to be exchanged to keep the semantics intact.

Figure 2 shows an extract of the Ontology of Intertextuality [2]. It describes a hierarchy of relational concepts and subconcepts. In addition, is specifies that the concept *inspire* is transitive, the concept *similar audience* is symmetric, and the concepts *retell* and *is-retold* are inverse to each other.

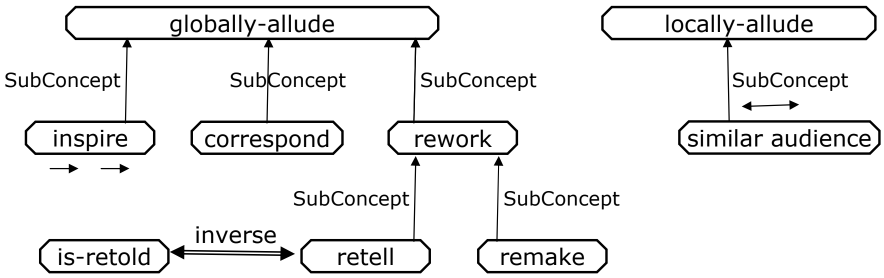

Figure 2: Extract from the Ontology of Intertextuality

6 Ontology-based Query Refinement

The integration of ontology knowledge into the EMMO model allows to refine user queries. Thus, a user can pose imprecise queries, which are refined by drawing inferences over ontological knowledge. There are two possible ways of incorporating ontological knowledge: either by explicitly extending the graph structure of an EMMO with additional associations or by implicitly extending the navigational operators in EMMA. The decision, which of the two options is preferable for a given situation depends on several factors, e.g. depth and size of the ontology or the size of an EMMO. A thorough evaluation of the influence of such criteria on the query performance for these two choices is subject of future work. The final goal is that we can choose the best solution under certain circumstances in a flexible way. We also plan to provide EMMA operators to change freely between these two representations. For example, in a distributed environment, before sending an EMMO, a user can *deflate* the EMMO to make it compact, whereas the recipient can fully *inflate* the EMMO again to achieve optimal query performance.

In the following, we will exemplify how the ontological knowledge about the subconcept hierarchy, as well as about inverse, transitive, and symmetric relational concepts can be used for query refinement. All the examples are based on the ontology structure illustrated in Fig. 2.

Integrating the Knowledge about the Subconcept Hierarchy: In our example, the subconcept hierarchy specifies that the concept *retell* is a subconcept of *rework*, which is again a subconcept of *globally-allude*. The explicit integration of this knowledge into the EMMO model can be realized by adding the associations $(l_{school} \stackrel{o_{rework}}{\longrightarrow} l_{dracula})$ and $(l_{school} \stackrel{o_{globally-allude}}{\longrightarrow} l_{dracula})$ to the EMMO "Dracula"(see Fig. 3).

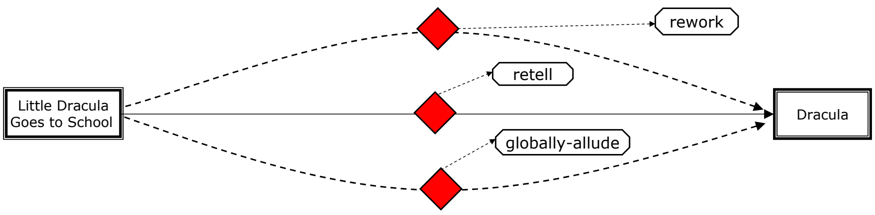

Figure 3: Integrating the knowledge about *retell*'s superconcepts

The other possibility of integrating the knowledge about the subconcept hierarchy is the implicit extension of EMMA's navigational queries by adding subconcepts as alternative branches. For example, the query $JumpRight(e_{dracula}, l_{school}, o_{rework}) = \emptyset$ can be expanded to search also for all subconcepts of *rework*, i.e. *retell* and *remake*:

$$JumpRight(e_{dracula}, l_{school}, o_{rework} \,|\, o_{retell} \,|\, o_{remake}) = \{l_{dracula}\}.$$

Thus, a user requesting all entities which were *reworked* by the book "Little Dracula Goes To School", receives the logical media part "Dracula" because the book *retells* the movie.

Integrating the Knowledge about Transitive Concepts: To integrate the knowledge that the concept *inspire* is a transitive concept into the EMMO model, we can either add the association $(l_{frank} \stackrel{o_{inspire}}{\longrightarrow} l_{dracula})$ to the EMMO "Dracula" (see Fig. 4) or expand the EMMA

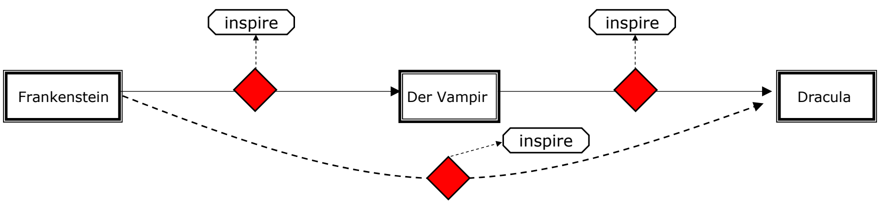

Figure 4: Integrating the knowledge that *inspire* is a transitive concept

query $JumpRight(e_{dracula}, l_{frank}, o_{inspire}) = \{l_{vampir}\}$ to the query

$$JumpRight(e_{dracula}, l_{frank}, o_{inspire}+) = \{l_{vampir}, l_{dracula}\}.$$

Integrating the Knowledge about Symmetric Concepts: To explicitly incorporate the knowledge about *similar audience* being a symmetric concept, we add the association $(l_{bite} \stackrel{o_{similar}}{\longrightarrow} l_{school})$ to the EMMO "Dracula" (see Fig. 5).

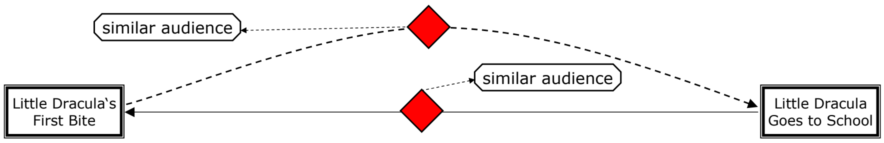

Figure 5: Integrating the knowledge that *similar audience* is a symmetric concept

As an example of the implicit integration into EMMA, we can expand the EMMA query $JumpRight(e_{dracula}, l_{school}, o_{similar}) = \emptyset$ to the expression

$$JumpRight(e_{dracula}, l_{school}, o_{similar} \,|\, o_{similar}-) = \{l_{bite}\}.$$

Integrating the Knowledge about Inverse Concepts: The fact that the concepts *retell* and *is-retold* are two inverse concepts can be either expressed by adding the association $(l_{dracula} \stackrel{o_{is-retold}}{\longrightarrow} l_{school})$ to the EMMO "Dracula" (see Fig. 5),

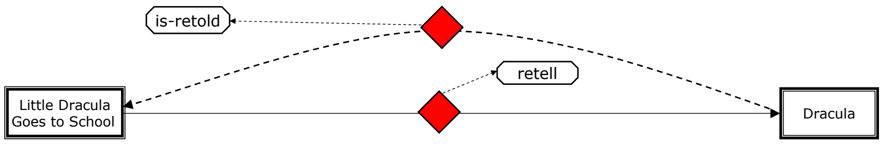

Figure 6: Integrating the knowledge that *retell* and *is-retold* are inverse concepts

or by expanding, for instance, the query $JumpRight(e_{dracula}, l_{dracula}, o_{is-retold}) = \emptyset$ to the query

$$JumpRight(e_{dracula}, l_{dracula}, o_{is-retold} \,|\, o_{retell}-) = \{l_{school}\}.$$

7 Conclusion

In this paper we have presented how we extend the EMMO model to provide a means to refine user queries by incorporating ontological knowledge. Currently, we are in the process of compiling a comprehensive set of use cases for the performance evaluation of explicit and implicit query refinement. Regarding the ontology construction, we currently use Protege-2000 to build the domain ontology, and import the resulting OWL [15] description into the EMMO environment. However, future work will also focus on the development of an ontology description language that is fully compatible with the EMMO model. This way we will be able to construct the domain ontology within the EMMO authoring environment so that we can offer advanced integrity checks for modeling and inference capabilities for query refinement. Finally, after finishing the implementation, we will also address research issues regarding the query optimization and the presentation of the query results.

References

[1] J. Ayars et al. Synchronized Multimedia Integration Language (SMIL 2.0). W3C Recommendation, World Wide Web Consortium (W3C), August 2001.

[2] M. Benari et al. Proposal for a Standard Ontology of Intertextualtiy. Public Deliverable Version 2.0, CULTOS Consortium and Project Planning, June 2003.

[3] D. Bogachev. TMPath – Revisited. Online Article, available under http://homepage.mac.com/dmitryv/TopicMaps/TMPath/TMPathRevisited.html, 2004.

[4] D. Brickely and R. Guha. Resource Description Framework (RDF) Vocabulary Description Language 1.0: RDF Schema. W3C Working Draft, World Wide Web Consortium (W3C), April 2002.

[5] J. Davies, R. Weeks, and U. Krohn. QuizRDF: Search technology for the Semantic Web. In *Proc. of the 37th Annual Hawaii International Conference on System Sciences (HICSS'04)*, Big Island, Hawaii, 2004.

[6] D. Fensel. *Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce*. Springer, Heidelberg, 2001.

[7] J. Ferraiolo, F. Jun, and D. Jackson. Scalable Vector Graphics (SVG) 1.1. W3C Recommendation, World Wide Web Consortium (W3C), January 2003.

[8] C. Flut et al. Spectacle. In *Towards the Semantic Web: Ontology-driven Knolwedge Managment*, West Sussex, UK, 2003.

[9] F. Frasincar et al. RAL: An Algebra for Querying RDF. In *Proc. of the Third International Conference on Web Information Systems Engineering (WISE 2000)*, Singapore, 2002.

[10] L. Garshol. Tolog 0.1. Ontopia Technical Report, Ontopia, 2003.

[11] ISO/IEC JTC 1/SC 34/WG 3. Information Technology – SGML Applications – Topic Maps. ISO/IEC International Standard 13250:2000, International Organization for Standardization/International Electrotechnical Commission (ISO/IEC), February 2000.

[12] G. Karvounarakis et al. RQL: A Declarative Query Language for RDF. In *Proc. of the 11th International World Wide Web Conference (WWW 2002)*, Honolulu, Hawaii, 2002.

[13] O. Lassila and R. Swick. Resource Description Framework (RDF) Model and Syntax Specification. W3C Recommendation, World Wide Web Consortium (W3C), February 1999.

[14] K. Schellner, U. Westermann, S. Zillner, and W. Klas. CULTOS: Towards a World-Wide Digital Collection of Exchangeable Units of Multimedia Content for Intertextual Studies . In *Proc. of the Conference on Distributed Multimedia Systems (DMS 2003)*, Miami, Florida, 2003.

[15] P. Schneider, P. Hayes, and I. Horrocks. OWL Web Ontology Language Semantics and Abstract Syntax. W3C Recommendation, World Wide Web Consortium (W3C), February 2004.

[16] Z. Zhang et al. An Ontology-based Method for Querying the Web Date. In *Proc. of the 17th International Conference on Advanced Information Networking and Applications(AINA'03)*, Xi'an, China, 2003.