Ontology-based Integration of OLAP and Information Retrieval

Torsten Priebe, Günther Pernul
Department of Information Systems, University of Regensburg, Germany
Email: {torsten.priebe, guenther.pernul}@wiwi.uni-regensburg.de

Abstract

This paper describes an ontology-based approach for building an enterprise knowledge portal that integrates OLAP and information retrieval functionality to access both structured data stored in a data warehouse and unstructured data in form of documents. We discuss how to perform global searches over these information sources. In addition, our approach provides adaptive searching by tracking the user context. When a user performs ad-hoc navigation in an OLAP report, the system will be able to use the query context information to also search for relevant documents.

1. Introduction

A major challenge of today’s information systems is to provide the user with the right information at the right time. This paper describes an approach for building an enterprise knowledge portal that integrates OLAP and information retrieval functionality to access both structured data stored in a data warehouse and unstructured data (i.e. documents) such as news articles, product descriptions, etc. We discuss how to perform global searches over these information sources through the portal. In addition, our approach provides adaptive searching by tracking the user context. When a user performs ad-hoc navigation in an OLAP report, the system will be able to use the query context information to also search for relevant documents. The approach is based on a global RDF (and RDF schema) [9, 10] based ontology that stores the metadata for the different resources.

The remainder of this paper is organized as follows: Section 2 discusses the overall architecture of the proposed system. The presented approach is based on a global ontology, which is being elaborated in section 3. The key idea of this paper, which is applied for global searching as well as for the adaptive/context-based searching, is a similarity-based information retrieval approach on metadata which is presented in section 4. Section 5 shortly discusses how the presented approach can be used to dynamically search for documents relevant to an OLAP query context. Section 6 presents related work and compares our approach to others in the literature. This paper presents work in progress; there are still open issues that need to be resolved. Section 7 discusses these issues and further future work.

2. Overall Architecture

The approach discussed in this paper is based on the idea of a Web-based portal system, integrating three information sources: an OLAP system for data warehouse access, a document management system (DMS) to access unstructured data in form of documents, and native portal content such as news articles. It is assumed that a commercial OLAP system with its own individual metadata set is used, which makes some type of metadata integration necessary. We propose to extend the regular data warehouse ETL (extract, transform, and load) process for this purpose. For the DMS, however, we assume a system that supports RDF-based metadata storage. This way it can directly use the central repository, the global ontology. For both cases the ontology provides the necessary information for mapping between different metadata constructs so that the system components can communicate using “their own language”.

Figure 1 shows the overall architecture of the proposed system. The extended ETL process feeds the data warehouse and keeps the ontology up-to-date. Within the portal platform three portal components (so-called portlets [9]) are responsible for the user interface: The se-
semantic search portlet provides the search user interface and queries the metadata, the OLAP portlet represents the OLAP user interface with ad-hoc navigation capabilities (i.e. drilling and slicing/dicing [2]), and the content portlet is used to display native portal content. As an example we use a portlet for news articles. The OLAP portlet communicates with an OLAP server software (with proprietary metadata). The DMS is responsible for controlling the access to the documents; it uses the global ontology for its metadata.

As a case study we use a mail-order retail company that sells various consumer goods via calling centers (based on the VMall demo data set provided with the MicroStrategy 7i product).

3. The key to integration: a global ontology

The heart of the architecture is the global ontology. It contains a global data model plus construct mapping information. Besides this schema information we also include instance data for business objects that are of enterprise-wide importance (i.e. the product portfolio, the customers, etc.). These can be used for document tagging and searching. Such business objects are usually found as dimension elements in a data warehouse, thus we can generate or update them also during the data warehouse ETL process.

Finally the resource metadata is part of the ontology. As mentioned before we assume the DMS to have an RDF-compatible interface and store its metadata directly in the ontology. What remains is the metadata for (predefined) OLAP reports, allowing the search engine to find them upon user request. We propose to use information like the report title plus the dimension elements shown on the report for this purpose. As this can be derived from the report definition, it is possible to automatically generate these metadata sets, either also in a similar load process.

This leads an RDF schema representation of the data model (i.e. a set of RDF classes and properties) plus instances (i.e. RDF descriptions) for the business objects and resources. It turns out that RDF schema diagrams become quite unreadable when using the regular RDF notation. Hence, like other authors we use some abbreviations. The double arrow in figure 2 means that the resource connected by one arrowhead has (i.e. is the domain of) a property with the given URI. The property's range is represented by the resource (which should be an rdfs:Class) connected by two arrowheads. In addition we use an open arrow for inheritance (rdfs:subClassOf, dashed for rdfs:subPropertyOf) in figure 3. We use the following namespace declarations:

\[
\begin{align*}
\text{ rdf } & = \text{http://www.w3.org/1999/02/22-rdf-syntax-ns#} \\
\text{ rdfs } & = \text{http://www.w3.org/2000/01/rdf-schema#} \\
\text{ s } & = \text{http://www.mycompany.com/portal/schema#} \\
\text{ o } & = \text{http://www.mycompany.com/portal/objects/} \\
\end{align*}
\]

3.1. Representing the OLAP dimensions

Our first step for building the ontology is to represent the OLAP dimensions using RDF(S). The OLAP data model of the VMall demo set consists of five dimensions: the product dimension (with items, subcategories, categories), the promotions dimension (about promotions and promotion types for specific items), the customer dimension (customers in a regional context), the geography dimension (employees, calling and distribution centers in a geographic context), and finally the time dimension. The RDF schema representation of these dimensions is shown on the right-hand side of figure 2. The time dimension is not included as it will be treated differently in our model. The left-hand side shows some instances of the dimension constructs, i.e. the individual products, the product categories and subcategories, the calling centers, the regions, etc. These are what we called enterprise-wide business objects. Of course there could be additional ones which are not derived from OLAP dimensions.

3.2. Resource metadata

We now have the business objects as a starting point. What, for simplification reasons, is not shown in figure 2 is that all the business object classes are subclasses of a class s:Object. What remains are the actual resources. These will be subclasses of a class s:Resource. We derive s:OLAPReport for (predefined) OLAP reports, s:News for news articles as native portal content, and finally s:Document for documents of the DMS.
Figure 3 shows the RDF schema for our resource metadata. What all resource classes have in common is an s:modified property for the last modification date and an s:author property for the responsible employee. Note that we derived an s:User class from the (OLAP originating) s:Employee class which, besides using the label “User” adds a s:userId property. Also note, that for OLAP reports the s:author property is renamed into s:owner as this is the terminology used by the proprietary OLAP metadata.

In addition for news articles the s:modified date is renamed into s:releaseDate, which goes along with a s:releaseTime property. Instead of using literals, date values are represented as anonymous instances of the s:Date class (which uses the rdf:value property for the actual date value in “yyyy/mm/dd” format).

All resources also have an s:about property associated with them. This can be used to describe the content of the resource by selecting related business objects. As you will easily see, for OLAP reports (which might have a constraint in the time dimension) this can also involve dates or date ranges. This is why s:Date and s:DateRange (which is modeled as two literal values representing the start and the end of the time period) are subclasses of s:Object. Note that the s:releaseDate of a news article is not only a subproperty of s:modified but also of s:about, as the release date of a news article certainly asserts something about the news content.

Figure 4 shows the metadata for two sample resources. Under a) you see the metadata for a news article about the opening of a new calling center in Atlanta. The s:author property points to the employee Michael Bates, release date and time are set to January 5, 2000 10:05:23.

Finally, an s:about states that the news article is related to the business object representing the “Atlanta” calling center. Note that the s:releaseDate property also involves an implicit s:about property.

Although information like the author and the last modification date can possibly be recorded automatically, useful metadata for resources like documents and news articles obviously depends on manual voluntary tagging by users using document or content management tools. Such systems should provide point-and-click interfaces for linking resources to business objects such as products or customers.

This is different for the metadata of the OLAP report shown under b) in figure 4. It is automatically derived from the report definition (i.e. the proprietary OLAP metadata). The report is owned by Thomas Smith, was last modified on January 14, 1999. In addition its content is related to the Freeplay Solar Radio, the Micro FM Radio, and the Shower Companion products. These are the dimension elements shown on the rows of the report (see figure 5). The columns show the different quarters of 1998 which are represented as date ranges.
4. Information retrieval on metadata

Now after we have a global metadata set in place, how do we actually search it for relevant resources? Let’s assume the user wishes to search for resources about the Freepay Solar Radio product that have the word “sales” in the title. The most straightforward search approach would be to simply perform an exact query on the RDF repository and return such (and only such) resources that fulfill the criteria specified as search constraints by the user. However, there are two problems. First, the metadata quality depends on the users’ tagging, which is a voluntary process. Second, we expect an enterprise metadata model to become quite complex. It is thus hard for users to build search queries that “perfectly” represent their information need.

For these reasons we suggest to review the classical information retrieval process [1]. The goal is to find documents (or more generally speaking, resources) that are relevant to an information need specified by a user. Usual IR queries (those who represent keyword-based full-text searches) are of a fuzzy nature. A query returns a ranked list of documents to the user showing most relevant ones first. Our goal is to find a similar way of searching metadata.

4.1 Similarity of metadata sets

We borrow our general search approach from information retrieval models such as the vector room model (VRM) [1]. The VRM is based on the similarity of document and query representations. Every document is represented as a vector of term frequencies (i.e., how frequently certain keywords occur in the documents). Another vector is created from the query keywords. The matching (and ranking) of the documents is done on the basis of the similarity of the individual document vectors and the query vector. Distance measures such as the cosine similarity are used.

Translating this into the world of semantic metadata means that we represent both the resources and queries as RDF metadata sets. As a user defines his query by constraining certain properties, this set of properties can be represented in RDF by an anonymous rdf:Description. The search can then be performed by calculating a distance measure (or match percentage) between the query and each individual resource description. Two axioms can easily be identified: Two identical descriptions have the distance 0 (100% match), two completely unrelated descriptions have the distance \( \infty \) (0% match). A good similarity measure, however, still has to be determined (and tested in a practical setting). The next section presents some preliminary ideas.

4.2 Matching and fuzzy queries

The above query would be represented as an anonymous RDF description with two properties as shown in figure 6.

![Figure 6. Sample query as RDF description](image)

A very simple preliminary matching would award resources that have the word “sales” in the title and that are related to (about) the Freepay Solar Radio product with a 100% match value. Resources with only one of the two properties would receive a 50% match value; resources with neither of the two, 0%.

However, with the above simple matching approach we would only find resources that have a direct relation to the Freepay Solar Radio product (i.e., an s:about property pointing to the corresponding business object). A resource about the audio product subcategory would have a 0% match, although it is (indirectly) related to the Freepay Solar Radio product, which belongs to the audio subcategory (i.e., it has an s:hasSubcategory property pointing to the corresponding subcategory object). We propose that such distant relations should also be found, but the match value should be less than for a direct relation (e.g., 50% for the above example and 25% for a distance of 3).

As we described in the last section we represent date information as dates and date ranges, resulting in anonymous RDF descriptions of type s:Date or s:DateRange. This has to be considered for the queries which should also find dates that lie within a date range or overlapping date ranges. With respect to the matching it needs to be defined whether an exact match (same date or date range) is “better” than just an overlap.

5. Utilizing the OLAP query context

As our matching approach is based on the similarity of RDF metadata sets, it is also possible to use a resources description itself as a query. A portlet displaying the OLAP report in figure 5 could provide a “Find Related” link for this purpose. It would lead to an implicit query using the metadata shown in figure 4.b) as search constraints.

On first sight this reveals the need for a weighting of individual property constraints. A modification date is certainly not as relevant for a search query as a property that constraints the content to being related to certain products. It would thus be desirable to assign weights to
property classes. In addition the user might want to dynamically weight certain properties on a per-query basis.

Nevertheless, with the weighting issue resolved, this allows searching for resources relevant to a currently displayed OLAP report. The clue is that this works not only for static predefined reports but also for ad-hoc OLAP queries that dynamically evolve through drilling and slicing/dicing. Also for such dynamic reports it is possible to automatically generate an RDF description that represents the content displayed in the report. Such a description can then be used to search for related other resources (such as news articles or other documents).

This is a unique feature of our approach; it provides adaptive searching by tracking the user context.

6. Related work

In the EU funded project GOAL (Geographic Information Online Analysis, INCO COPERNICUS project no. 977071) [4] in which the authors were involved, the integration of data warehouse (or more precisely OLAP) technology and geographical information systems (GIS) was analyzed. The basic idea behind such integration is that a geographical OLAP dimension can be mapped to GIS objects such that maps can be used to navigate through OLAP data.

Like us, [5] address the integration of unstructured (or semi-structured) documents with structured OLAP data. They treat OLAP cubes as “documents” stored in a digital library like repository and use manually created metadata to link them to related documents. Our approach goes a step further, the OLAP cubes are not treated as a black box, but the navigation inside the cubes (the above mentioned query context) is also considered for retrieving related documents.

Such context-based information retrieval has been studied by [3] who propose an architecture for finding documents relevant to the context the user is in. They propose to include plug-ins in client applications that would communicate the user’s working context to an information retrieval engine. However, most application programs will not easily allow the integration of plug-ins. In addition, the identification of a user context and its translation to an IR query in a totally generic way seems problematic. The use of an MIS (or OLAP) query context is explicitly mentioned as promising by [3] but not elaborated in more detail.

Information retrieval on the Semantic Web is also discussed by [6]. They propose a hybrid approach of metadata- and full-text-based searching. Actually, combining our metadata-based approach with regular key-word based IR would be quite interesting and will be part of our future work.

7. Conclusions and future work

We have presented a retrieval approach to globally search for resources from different information sources. This approach allows fuzzy queries over resource metadata. It should be noted, that although we presented the query approach using a concrete sample metadata model, the ideas are independent of any specific model. They should hold for basically any RDF based metadata.

However, there are still some open issues: We need a proper distance measure to quantify the similarity of two RDF descriptions. For example, we need a weighting of property classes; this is what we are currently working on. In parallel a prototype system is being built to evaluate the ideas. Future work will deal with combining our retrieval approach with full-text search capabilities and integrating other additional information sources, such as operative applications (e.g. a CRM system) into the portal.

References