Taking Enterprise Search to the Next Level

Kay-Uwe Schmidt
SAP Research
Vincenz-Prießnitz-Straße 1
76131 Karlsruhe, Germany
kay-uwe.schmidt@sap.com

Daniel Oberle
SAP Research
Vincenz-Prießnitz-Straße 1
76131 Karlsruhe, Germany
d.oberle@sap.com

Klaus Deissner
SAP AG
Hasso-Plattner-Ring 1
69190 Walldorf, Germany
klaus.deissner@sap.com

ABSTRACT

Enterprise search is vital for today's enterprises. The ongoing growth of information in enterprises demands new solutions for finding relevant information in the information space. The personalization of search results is one promising approach in solving this challenge. Additionally, ontologies stoke expectations of easing the information integration process for federated search results by their formal and declarative nature. In this poster we present our novel approach of an ontology-based personalized enterprise search. We introduce an ontology-based federation layer for bridging the heterogeneity of the different knowledge sources in an enterprise.

1. INTRODUCTION

Consistent and efficient search in a company's ever increasing knowledge sources becomes increasingly important for todays enterprise information workers. However, the heterogeneity of such knowledge sources makes this task very difficult compared to conventional Web serach. In [1] the differences and challenges of enterprise search are summarized. So one major difference is the diversity of content sources and formats in enterprises as well as the need for personalized search based on the current context of the information worker. The issue of personalization is also emphasized by [2]. Hawking points out that exploiting search context within enterprise searches is one of the key IR research problems in the area of enterprise search.

Consequently, enterprise software vendors like SAP completed their portfolio by *enterprise search* engines[1]. In parallel third party vendors specialized in search developed enterprise search solutions like FAST[2] or Autonomy[3]. According to [3] FAST and Autonomy still are the "800 pound gorillas" in the market. Enterprise search engines are federated search engines which propagate user queries to several search service providers, i.e., business applications and data sources such as SAP ERP (Enterprise Resource Planning), Business Intelligence applications or Knowledge Management systems, via connectors. A connector is a software module encapsulating the access characteristics of a search service provider. A federation layer then presents the individual results in an integrated way to the user via arbitrary information channels.

[1]http://www.sap.com/solutions/informationworkers/enterprisesearch/index.epx
[2]http://www.fastsearch.com
[3]http://www.autonomy.com

We focus on several typical shortcomings of the federation layer and improve upon them considerably. We accomplish this by introducing an ontology-based federation layer into the enterprise search architecture. Further more we leverage the expressive power and reasoning capabilities of F-logic in order to have an highly dynamic and easy to maintain approach. By introducing the ontology-based federation layer we set the stage for a personalized and dynamic categorized enterprise search system.

2. SHORTCOMINGS OF THE FEDERATION LAYER

Static categorization of search results: Current enterprise search engines (e.g., the SAP Enterprise Search solution) only provide the users with a static set of categories to select from, independent from what the user's search term is. There is no support for dynamic categorization. By dynamic categorization we mean the process of assigning search results to categories of different business domains, at run time, possibly with the consideration of the user's context. **Missing flexibility:** A major challenge of the federation layer is to harmonize the heterogeneous results provided by the connectors. Thus, realizing the federation layer requires building a data model for representing and processing the results delivered by the individual connectors. Current state-of-the-art follows a model-driven approach, i.e., the model is drawn with CASE tools at design-time and subsequently transformed to code. As a consequence, fundamental changes in the model require reengineering of the model or reconfiguration. We improve upon this situation considerably by decoupling the connectors' models from the business domains and retain the link between both with a maximum of flexibility. **Need for optimization:** If the above decoupling between the connectors' models and the business domains is in place, it can be exploited for optimizing the actual queries to the search service providers. **Personalization:** Typical enterprise search engines do not provide means for considering the user's context. Result visualizations look alike even if the users are in completely different roles. Search results should be displayed according to the user's context, i.e., depending on whether the user is a sales person, consultant, etc.

3. DESIGN

Our solution remedies the four shortcomings by introducing a generic, ontology-based federation layer. The principal idea is to separate the technical from the business

domain models (in our case represented via ontologies) by declarative mappings (represented in the ontology language, too). Given this flexible, declarative decoupling, we realize dynamic categorization and optimization by querying the mappings. The ontology-based solution further allows for a personalized view on the search results based on the user's context (sales, consultant, publishing, etc.).

From the shortcomings introduced in Section 2, we can derive the following primary requirements. Primary requirements are ones that follow directly from analysing the problem space: **Flexible representation of business and technical domain(s):** The business and technical domains must be represented flexibly. Each business domain consists of its own concepts and relations. The business domains (sales, publishing, etc.) have to be represented in a flexible way because changes are likely and required in different installations. **Link the business domains to technical domains:** The entities (concepts and relations) of the business domains (e.g. [product], [customer], [sales order], etc. in the sales domain) have to be linked to the entities of the search service providers, i.e., the technical domains (business objects, tables, etc.). **Consider the user context:** The user context consists of dynamic and static context information. Dynamic information comprises the location, the working context [4, 5] or the workflow currently processed by the user. Static user context information comprises the role of a user in an enterprise. An enterprise search application should at least consider the static parts of a user's context, e.g., if the user is a sales person, the federation layer should only take into account results which are relevant for the sales domain (e.g., [product] and [customer] categories, or [ordered by]). Entities of other domains should not be excluded but clearly marked or separated.

4. ARCHITECTURE

Figure 1 sketches the generic architecture. Basically, an ontology-based federation layer has to be introduced. Search providers, connectors and web application could in principle be reused from an existing enterprise search engine. The ontology-based federation layer cares for data representation, mapping and flow via ontologies. A federation controller cares for the control flow between the Web application and controllers. An ontology editor can be used at design time to maintain ontologies and mappings.

Our primary architectural decision is to use ontologies in the solution because they meet our requirements. An ontology is a declarative specification of concepts and their relations in a logic-based language avoiding ambiguities and providing the basis for querying and reasoning tasks. Unlike "model and forget" (e.g., UML or ERM) and "model and transform" (e.g., MDA) approaches, ontology tooling comes with an API that allows to create, change, query and reason with ontologies at runtime. There will be the following three types of ontologies: Business Domain, Technical and User Info Ontologies (see Figure 1).

Business Domain Ontologies There will be n ontologies for capturing n business domains. Each business domain ontology is a pure conceptual model whose concepts and relations reflect the terminology of the information worker. The business domain ontologies will be populated via mappings from the instances of the technical info ontologies.

Technical Ontologies For m search service providers there will be m technical ontologies. Each of them is a

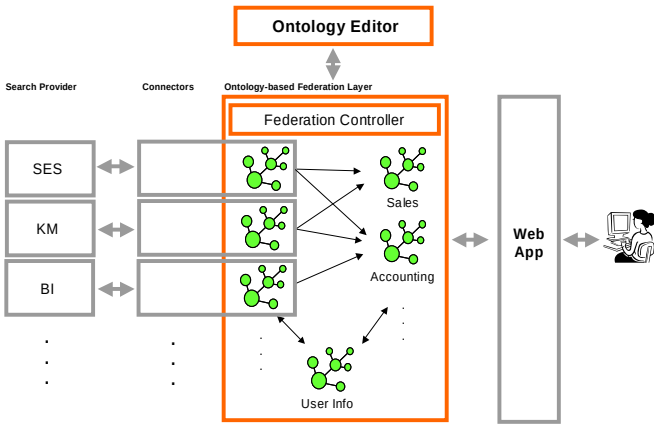

Figure 1: General architecture. The ontology-based federation layer is the contribution of our approach.

simple representation of the search service providers' data structures. In the case of SAP R/3, this would be business objects and their attributes, in the case of a knowledge management application this would be tables and their columns and so on. The search results will be represented by instances in the respective ontologies and interlinked with the respective search instance (cf. user info ontology). Furthermore, complex $n : m$ ontology-to-ontology mappings will be defined between business and technical ontologies.

User Info Ontology There will be one ontology for representing user information at run-time such as the user context, the search terms, or timestamps. This user info ontology is also a link between the business domain ontologies and technical information ontologies. In our running example, this ontology would feature an instance of a concept `[Search]` with information about who issued the search *Daniel*, when, as well as which search terms were given. Also, this instance would be interlinked to the search results represented in the technical ontologies. For example, the `[Search]` instance would be interlinked with instances of the `BUS1178` concept if *champagne* has been found in this business object in SAP R/3 via SES.

5. REFERENCES

[1] Rajat Mukherjee and Jianchang Mao. Enterprise search: Tough stuff. *ACM Queue*, 2(2):36–46, 2004.

[2] David Hawking. Challenges in enterprise search. In Klaus-Dieter Schewe and Hugh E. Williams, editors, *ADC*, volume 27 of *CRPIT*, pages 15–24. Australian Computer Society, 2004.

[3] Mark Bennett. Enterprise search engine vendors in 2007. Enterprise Search Newsletter Volume 4 Number 1, New Idea engineering Inc., 2007.

[4] Pavol Návrat and Tomas Taraba. Context search. In *Web Intelligence/IAT Workshops*, pages 99–102. IEEE, 2007.

[5] Reiner Kraft, Chi-Chao Chang, Farzin Maghoul, and Ravi Kumar. Searching with context. In Les Carr, David De Roure, Arun Iyengar, Carole A. Goble, and Michael Dahlin, editors, *WWW*, pages 477–486. ACM, 2006.