IDEA GROUP PUBLISHING

701 E. Chocolate Avenue, Suite 200, Hershey PA 17033, USA
Tel: 717/533-8845; Fax 717/533-8661; URL-http://www.idea-group.com

ITP4812

Enhanced Knowledge Warehouse in the Semantic Web

Krzysztof Wecel and Witold Abramowicz
Department of Management Information Systems
The Poznan University of Economics
Al. Niepodleglosci 10
60-967 Poznan, Poland
{K.Wecel,W.Abramowicz}@kie.ae.poznan.pl

Pawel Jan Kalczynski
Department of Information Systems
University of Toledo
2801 West Bancroft St.
Toledo, OH 43606, USA
Pawel.Kalczynski@utoledo.edu

ABSTRACT

This paper presents how the enhanced Data Warehouse system was remodeled in order to transform it from a closed solution to an open web-services-based system called enhanced Knowledge Warehouse. We describe the modeling framework used, the Web Service Modeling Framework (WSMF). Further we analyze eKW as a web service and show how eKW conforms to eight layers of functionality in web services. Finally, we show how eKW could be embedded and used in the Semantic Web, and what work is required to achieve a fully-fledged system.

INTRODUCTION

Until now the Web focused on publishing information that is readable primarily for humans. However, recently more and more attention has been paid on processing information automatically by computers. To achieve this goal sophisticated systems are designed. They use various techniques of Artificial Intelligence, e.g. shallow text processing.

Tim Berners-Lee suggested another solution – to create the Web so that it will be easily processable by machines. Such a web is called the Semantic Web [Berners-Lee, 2001].

Another issue is making application accessible through the Web. The ultimate vision is that of the Web as of a distributed computation device.

According to the IBM web service tutorial, "web services are a new breed of Web application. They are self-contained, self-describing, modular applications that can be published, located, and invoked across the Web." [WSCA2001].

The idea we present in this paper was previously called the *enhanced Data Warehouse (eDW)* [AbrKalWec, 2002]. eDW was primarily designed as a closed system. Only users of a particular data warehouse could take advantages of this solution. Moreover, eDW was based only on internal modules without taking advantages of other systems. According to the recently observed trends, we decided to re-engineer the architecture of the eDW system.

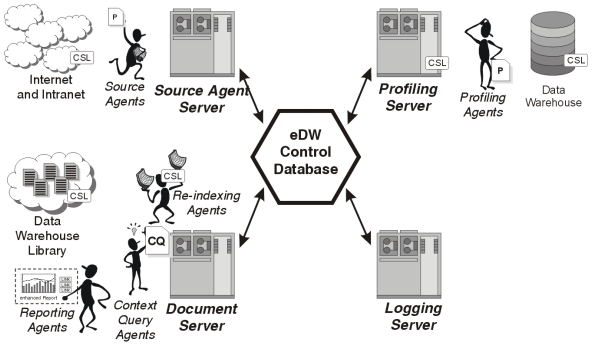
enhanced Data Warehouse

eDW is an agent-based system that allows the automatic filtering of information from the Web to the data warehouse and automatic retrieval through the data warehouse [AbrKalWec, 2002]. The overview of the system is presented in Figure 1.

In the original concept of eDW the Profiling Server was used to discover the information needs of data warehouses and to store these needs as profiles [AbrKalWec, 2001a]. The profiles were used by Source Agent Server to filter relevant documents from the Web, and store them in the Data Warehouse Library [AbrKalWec, 2001d]. Documents were accessible to users through the Document Server that responded to context queries, which represent the temporary information needs of users.

The weakest part of eDW is the component responsible for matching documents with information needs. We need a better integration with external information providers in order to serve user needs better.

Figure 1. Overview of the Enhanced Data Warehouse



If we want to take advantages of the Semantic Web to further develop eDW, we have to make it accessible through the Web. eDW is modularized, so we can easily transform each module to a self-contained service.

In the Semantic Web eDW should be used as a source of knowledge, hence the name *enhanced Knowledge Warehouse (eKW)*.

To model eDW in terms of the new architecture we need an appropriate framework. We decided to use a full-fledged Web Services Modeling Framework, which we will briefly describe below.

THE WEB SERVICE MODELING FRAMEWORK

The Web Service Modeling Framework (WSMF) is based on two principles [Fensel, 2002]:

- strong *de-coupling* of the various components that realize business application
- strong *mediation* service enabling anybody to exchange information with anybody else.

The first principle requires that any complex service should be decomposed into a number of smaller modules. Therefore, eDW was decomposed into many services that can act independently. These are:

a) *Library Service*, former Data Warehouse Library (DWL), derived from the Document Server
b) *Profiling Service*, evolved from the Profiling Server
c) *Filtering Service*, previously the Source Agents Server
d) *Indexing Service*, derived from back-office part of the Document Server
e) *Reporting Service*, derived from front-office part of the Document Server.

The second principle states how to connect the de-coupled services together. This is achieved by mediation of different vocabularies as well as by different interaction styles. The approach presented in WSMF is to provide a scalable interoperability among services.

The Web Service Modeling Framework consists of four main elements: ontologies, goal repositories, web services, and mediators.

Ontologies

Ontologies are considered a key enabling technology for the Semantic Web. Thanks to them, it is possible to represent knowledge that is understandable for humans and readable for computers [Fensel, 2001; Gruber, 1995].

In WSMF, ontology provides the terminology that is used by all other WSMF elements.

Ontologies define the following:

- *formal semantics*, allowing information processing by a computer
- *real-world semantics*, linking the machine-processable content with certain meaning for a human user.

In eKW the ontology occupies a central place. Because eKW is primarily designed to fulfill user needs, these needs should be specifically well described. Both profiles and context queries are expressed in terms of ontologies. So far, there is no agreed ontology, specific for eKW. This is a subject to research further.

Goal Repositories

A *goal* is an objective that a client may achieve while contacting the web service. A goal specification consists of two elements [Fensel, 2002]:

- pre-conditions, what a service expects as an input
- post-conditions, what a service returns.

It is advised that goal specifications are kept separately from actual web service descriptions, because one service can help in achieving different goals and one goal can be achieved by employing different services.

What is stressed in WSMF is that the goal should be precisely described. This is achieved by utilizing ontologies in the goal specification.

eKW ontology is utilized to build profiles and context queries. In a natural way they are well-defined pre-condition goal specifications. Profiles specify what documents should be returned by the Filtering Service, and context queries specify what documents should be returned by the Library Service, e.g. documents for a given period, for a given subject, by a selected author, of a specified length, etc.

In eKW we can distinguish the following goal repositories, presented in Table 1.

Web Service

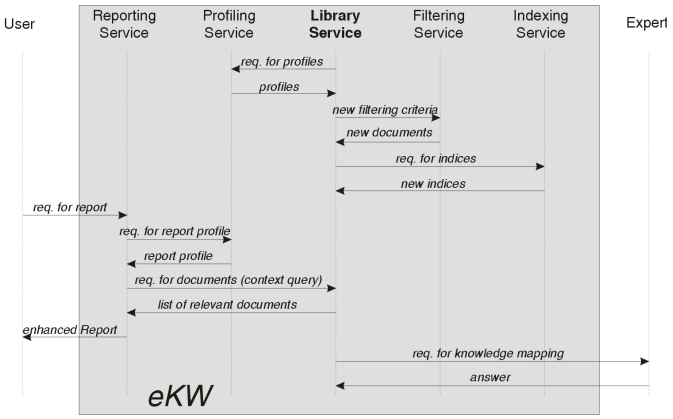
In a general sense, a web service is complex when it is composed of other services. However, in the WSMF there is a specific distinction between elementary and complex services. The criterion that matters is a complexity of service description (its interface).

According to WSMF, eKW is considered a simple web service, although it consists of sub-services (see Figure 2). Private processes are hidden from external users, and only external aspects (interface) of the service should be described.

Table 1. Goal repositories in eKW

Service	pre-conditions	post-conditions
Profiling	- request for profiles	- profiles
Library	- context query	- relevant documents
Filtering	- profiles	- relevant documents
Reporting	- request for reports	- *enhanced Report*
Indexing	- web document	- indices of this document

Figure 2. Message exchange sequence in eKW



There are some issues that should be discussed in more detail. Web service descriptions, like goals, contain pre-conditions and post-conditions. These conditions can be linked directly or indirectly (via a mediator) to goal conditions. In the second case a web service can strengthen a pre-condition or weaken a post-condition of a goal, because not all results of this web service fulfill the goal descriptions. When applied to the Library Service of eKW, these considerations remind the famous precision-recall tradeoff. The more goal conditions, the more accurate results, and the less number of documents returned.

In eKW a concurrent service binding method should be introduced. Then it would be possible to choose different web services for the same task, e.g. in eKW we can use different specialized indexing services. For example, if our service does not know how to index PDF files, we can use another service.

For each service in eKW we can declare an *invoked web service proxy*. This is useful when one web service may invoke other web services to provide its service. For example, the Reporting Service has to invoke the Profiling Service in order to obtain the profile of the user report. Proxy allows referring the web service, without defining, which web service will be invoked. The binding takes place during runtime. To continue our example, different profiling services may be called for different types of reports.

Special attention should be paid to errors. Complex services may use specialized *error ports*. If a web service performs long transactions, it should inform the service requester about it. For example, the Filtering Service requires time to find appropriate documents on the Web. Sometimes web services include concurrent data input and output streams. For example, the Reporting Service may negotiate input parameters such as specifying more accurate time constraints, requesting more/less documents in the list with the Library Service. If the Library Service does not return any documents that meet criteria specified in the context query, the Reporting Service may negotiate with the user different input parameters (weaker conditions).

Mediator

The concept of mediator was developed in heterogeneous and distributed information systems. Mediator translates user queries into subqueries on different information sources and integrates the sub-answers [Wiederhold, 1992].

WSMF distinguishes different types of mediation: mediation of data structures, business logics, message exchange protocols, and dynamic service invocation [Fensel, 2002].

Actually, the whole eKW is a mediator. It allows mediation between a data warehouse and the Web. Therefore, other types of mediations will not be further analyzed.

Table 2. Intentions in eKW in relation to document types

Document	Intentions
profile	- create new profile - store profile in a database - search for documents (request for filtering)
context query	- create new query - search for documents (request for retrieval)
indices	- create new indices (for a given document) - store indices in a database as semantic descriptions of documents
web document	- filter from the Web - store documents in a library - retrieve from a library - deliver to user
list of web documents	- compose the list - display the list to a user

Data could be mediated by direct mapping, e.g. using XSL-T rules for XML documents. However, this technique proves inefficient. Omelayenko and Fensel suggest a layered integration architecture, in which the mediation of data structures can be solved in two steps [Omelayenko, 2001]:

intermediate data model and three sub-steps: extract, map, rewrite; to cope with different syntactical standards

intermediate conceptualization (i.e. an Ontology); to cope with the number of mappings.

Mediation could be a web service itself, hence the idea of eKW as a web service. There is one interface for the Web, but many interfaces for different data warehouse solutions. Also, other management information systems may be taken into account.

EKW AS A WEB SERVICE

To consider eKW as a web service, we need to analyze web services in more detail. [Fensel, 2002] and [Bussler, 2001] identified eight layers necessary to achieve automatic web service composition into complex services. They are discussed further.

Document Types

First, we have to distinguish and define different document types that will be exchanged within the eKW system.

Document types describe the content of business documents, and are defined in terms of elements. They are instantiated when the service requester and provider exchange data.

Document types in eKW:

- *profile*, representation of relatively constant user information needs
- *context query*, unlike the profile, it defines temporary user information needs
- *web documents* (e.g. HTML, XML, PDF), document retrieved from the Web
- *indices*, representations of web documents that can be easily matched with a profile or a context query
- *list of web documents*, system's response to a profile or a context query.

Semantics

Documents should be semantically correct. This ensures that they are properly interpreted.

One of the most popular ways to conform to semantics is to use *ontologies*, which provide a means for defining the concepts of the exchanged data. Documents may refer to the ontology concepts, therefore the system knows proper element values. This ensures consistency and allows the same interpretation by all participants of the data exchange process.

Originally the semantic layer in eDW was provided by the Common Semantic Layer (CSL) [AbrKalWec, 2001c]. It was defined as a plain list of concepts used to index documents. The main drawback of this solution was the lack of relations between terms. Moreover CSL was only used to index documents and not to represent any knowledge in the system. In eKW we introduce ontologies to express semantics.

When talking about exchange of documents, we also have to define the intent of the exchange. Details are presented in Table 2.

Process Definition

When implementing web services, it is important to define the business message exchange sequence. The business logic of eKW is presented in Figure 2. The central module of eKW is the Library Service (formerly Data Warehouse Library). The first step is to find out what kind of information should be collected, therefore the Library Service consults the Profiling Service to get some representation of information needs. Then it can formulate new filtering criteria and pass them to the Filtering Service. When new documents that meet the criteria are found on the Internet, they are sent to the Library Service. In order to organize them, the Library Service sends documents to the Indexing Service, and receives new indices (both subject and temporal indices [AbrKalWec, 2001b]). Thus documents are ready for retrieval. The second block of processes starts with a request for the report submitted by a user. The user expects to get a data warehouse report accompanied by a list of documents relevant to this report. The request is handled by the Reporting Service. In order to retrieve relevant documents, the profile of the report is required. This profile can be obtained from the Profiling Service and it describes a report in terms of the ontology that allows matching documents against the report. Based on the profile, time constraints of the report, and some sophisticated algorithms, the Reporting Service builds an appropriate context query and submits it to the Library Service. In response, the Library Service returns documents. Finally, the Reporting Service prepares the *enhanced Report*. The final block of processes allows improving the knowledge of Library Service by consulting external experts. When some documents cannot be found automatically, an expert can suggest the matching and changes in ontologies.

Other layers

Transport Binding. There are plenty of data transport mechanisms available at the moment. The most popular are HTTP and HTTPS. It is also possible to use FTP or even SMTP when exchange is asynchronous. Prior to the exchange of data, the service requester and provider have to agree on the protocol. eKW will not use any sophisticated mechanism for SOAP, and so HTTP was considered the most appropriate.

Exchange Sequence Definition. Due to the unreliability of networks, service providers have to define a sequence of acknowledgment messages. Each message should be transmitted only once. This is a technical issue and it will not be addressed separately in eKW.

Security. Security is also a technical issue, and eKW will utilize standardized solutions. These solutions should provide encryption to ensure privacy and also signing to ensure non-repudiation.

Syntax. The most popular syntax is XML, and it was selected for eKW.

Trading Partner Specific Configuration. Each service requester or provider has different business logic. When partners want to cooperate, they have to start with some adjustments of these logics. Interaction should be formalized when using the web services.

ROLE OF EKW IN THE SEMANTIC WEB

The Semantic Web will provide access to heterogeneous and distributed information, enabling software products to mediate between user needs and the information sources available. On the other hand, the Web is a collection of information, and there are no means to process

this information. SWWS (Semantic Web-enabled Web Services) is a combination of web services together with the Semantic Web. As we have stated earlier in this paper, the main role of eKW in the Semantic Web is mediation. eKW is a kind of data mediator that employs ontologies as a conceptualization layer. This implies that one of the most important things that should be developed within eKW are ontologies. The first phase in the evolution of the Semantic Web will probably be to develop decentralized and adaptive ontologies [Kim, 2002]. Business related ontologies should be developed first. The necessary mediation between different information systems could be then carried out based on the ontologies. The use of ontologies would also allow better representation of user information needs. Because our original system was based on agents we decided to implement eKW as an agent-based system. New directions of research show that information agents together with ontologies can provide breakthrough technologies for Web applications. One of the most important languages for eKW is DAML-S (DARPA Agent Markup Language with ontology for Services). It is the ontology for services, and should make it possible to discover, invoke, compose, and monitor Web resources, which offer particular services and have particular properties. DAML-S could be then used as the service profile for advertising services.

CONCLUSIONS AND FURTHER WORK

This paper presented how enhanced Knowledge Warehouse was modeled to conform to a new way of building applications, namely to the web services. We were motivated by the incentives offered by the Semantic Web. Also, the idea of ontologies seems to be very convincing. This paper showed how the terminology from WSMF is utilized to model eKW as a Web service.

First of all, according to WSMF, we de-coupled our original system into separate web services. Those web services were then analyzed in terms of the eight layers of the web services functionality. This showed that only few layers require special treatment in eKW. We did not propose any formal notations.

In future work, the following languages will be useful to formally describe eKW in Semantic Web enabled Web Services: WSFL (*Web Services Flow Language*) [Leymann, 2001], a foundation for WSMFDAML-S (DARPA Agent Markup Language with ontology for Services) as web-based *syntax* [Ankolenkar, 2001] PSL (*Process Specification Language*) as a formal *semantics* [Schlenoff, 2000].

The Web becomes a global platform where organizations communicate among each other to exchange value-added services. The main service offered by eKW on this platform is to deliver information relevant to the user activities in a given context.

REFERENCES

[AbrKalWec, 2001a] W. Abramowicz, P.J. KalczyDski, K. W ę cel, „Profiling the Data Warehouse for Information Filtering", *Managing Information Technology in a Global Economy*, Proceedings of IRMA 2001 International Conference, Toronto 2001, pp. 810-814.

[AbrKalWec, 2001b] W. Abramowicz, P.J. KalczyDski, K. W ę cel, „Time Consistency among Structured and Unstructured Contents in the Data Warehouse", *Managing Information Technology in a Global Economy*, Proceedings of IRMA 2001 International Conference, Toronto 2001, pp. 815-818.

[AbrKalWec, 2001c] W. Abramowicz, P.J. KalczyDski, K. W ę cel, "Common Semantic Layer to Support Integration of the Data Warehouse and the Web", Human Computer Interaction – HCI 2001, Sopot.

[AbrKalWec, 2001d] W. Abramowicz, P.J. KalczyDski, K. W ę cel, "Information Ants to Explore Internet Sources of Business Information", *Proc.* of *ISCA 14th International Conference on Computer Applications in Industry and Engineering - CAINE2001*, Las Vegas, pp. 134-137.

[AbrKalWec, 2002] W. Abramowicz, P.J. KalczyDski, K. W ę cel, *Filtering the Web to Feed Data Warehouses*, Springer-Verlag London, UK, July 2002, 280 pp.

[Ankolenkar, 2001] A. Ankolenkar et al., DAML-S: Semantic Markup For Web Services, http://www.daml.org/services/ daml-s/2001/10/daml-s.html.

[Berners-Lee, 2001] Tim Berners-Lee, James Hendler, Ora Lasilla, „The Semantic Web", *Scientific American*, May 2001, http://www.scientificamerican.com/2001/0501issue/ 0501berners-lee.html.

[Bussler, 2001] C. Bussler, "B2B Protocol Standards and Their Role in Semantic B2B Integration Engines", *IEEE Data Engineering*, 24(1), 2001.

[Fensel, 2001] D. Fensel, Ontologies: Silver Bullet for Knowledge Management and Electronic Commerce, Springer-Verlag, Berlin, 2001

[Fensel, 2002] D. Fensel, C. Bussler, *The Web Service Modeling Framework*, Report Number IR-493, Vrije Universiteit Amsterdam, February 2002.

[Gruber, 1995] T.R. Gruber, "Toward principles for the design of ontologies used for knowledge sharing", *Int. J. Hum. Comp. Stud. 43*, 5/6, 1995, pp. 907-928.

[Kim, 2002] H. Kim, "Predicting How Ontologies for the Semantic Web Will Evolve", *Communications of the ACM*, 45(2):48-54.

[Leymann, 2001] F. Leymann, Web Services Flow Language (WSFL 1.0), May 2001, http://www-3.ibm.com/software/ solutions/webservices/pdf/WSFL.pdf.

[Omelayenko, 2001] B. Omelayenko, D. Fensel, "A Two-Layered Integration Approach for Product Information in B2B E-commerce", In *Proc. of the Second International Conference on Electronic Commerce and Web Technologies (EC-WEB 2001)*, Munich, Germany, September 2001.

[Schlenoff, 2000] C. Schlenoff, M. Gruninger, F. Tissot, J. Valois, J. Lubell, J. Lee, The Process Specification Language (PSL): Overview and Version 1.0 Specification, NISTIR 6459, NIST, Gaithersburg, MD (2000), http://www.mel.nist.gov/psl/pubs/PSL1.0/paper.doc.

[Wiederhold, 1992] G. Wiederhold, "Mediators in the Architecture of Future Information Systems", IEEE Computer, 25(3):38-49, 1992.

[WSCA2001] IBM Web Services Tutorial, http://www-3.ibm.com/software/solutions/webservices/pdf/WSCA.pdfFigure 2. Message exchange sequence in eKW.