Semantic Business Process Repository

Zhilei Ma[1], Branimir Wetzstein[1], Darko Anicic[2], Stijn Heymans[2], Frank Leymann[1]

[1]Institute of Architecture of Application Systems (IAAS)
University of Stuttgart, Germany
{firstname.lastname}@iaas.uni-stuttgart.de
[2]Digital Enterprise Research Institute (DERI)
University of Innsbruck, Austria
{firstname.lastname}@deri.org

Abstract. Semantic Business Process Management (SBPM) utilizes semantic technologies to achieve more automation throughout the BPM lifecycle. An integral part of the SBPM infrastructure is a semantic business process repository, which is used for storage and management of business process modeling artifacts. As in SBPM business process models are based on process ontologies, the semantic business process repository has additional requirements towards support of reasoning and querying capabilities. In this paper, we first describe the functionalities the semantic business process repository has to provide. We then introduce a solution based on the Integrated Rule Inference System (IRIS) on top of a relational database for realizing the storage mechanism and query processing. Finally, we present the overall architecture of the semantic business process repository.

Keywords: Business Process Management (BPM), Business Process Repository, Semantic Business Process Management (SBPM), Semantic Business Process Repository, Ontologies, Reasoning

1 Introduction

The globalization of the economy and the ongoing change of the market situation challenge corporations to adapt their business processes in an agile manner to satisfy the emerging requirements on the market and stay competitive against their competitors. Business Process Management (BPM) is the approach to manage the execution of IT-supported business processes from a business expert's point of view rather than from a technical perspective [SF03]. However, currently businesses have still very incomplete knowledge of and very incomplete and delayed control over their process spaces. Semantic Business Process Management (SBPM) extends the BPM approach by adopting semantic web and semantic web service technologies to bridge the gap between business and IT worlds [HLD+05].

In both BPM and SBPM business processes play a central role. As business processes manifest the business knowledge and logics of a corporation and normally more than one person or organization with different expertise and in different geographic locations are involved in management of business processes, it is necessary to estab-

In: M. Hepp, K. Hinkelmann, D. Karagiannis, R. Klein, N. Stojanovic (eds.): Semantic Business Process and Product Lifecycle Management. Proceedings of the Workshop SBPM 2007, Innsbruck, April 7, 2007, CEUR Workshop Proceedings, ISSN 1613-0073, online CEUR-WS.org/Vol-251/

lish a business process repository within the corporation for effective sharing of valuable business knowledge. Furthermore, business users tend to reuse existing business process artifacts during process modeling, so that they are able to adapt the business processes in a more agile manner. However, as the number of business processes increases, it is difficult for them to manage the process models by themselves and to find the required business process information effectively. A business process repository helps business users by providing a systematic way to manage and obtain information on business processes.

In SBPM, business process models are based on process ontologies and make use of other ontologies, such as organizational ontology and semantic web service ontology [HR07]. The business process repository has to be able to cope with these ontological descriptions when storing and retrieving process models, and in particular support efficient querying and reasoning capabilities based on the ontology formalism used. In order to distinguish from traditional business process repository technology, we call this kind of repository a semantic business process repository.

In this paper, we first analyze in section 2 the functional requirements on the semantic business process repository. We describe what kind of functionality the semantic business process repository should offer to its clients, which is primarily a process modeling tool. As a main issue, we identify the integration of a reasoner with the storage mechanism for query processing. In section 3, we then introduce a solution for data storage and query answering based on the Integrated Rule Inference System (IRIS[1]) on top of a relational database. Finally, in section 4, we describe the overall architecture of the semantic business process repository.

2 Requirements Analysis

In general, a repository is a shared database of information about engineered artifacts produced or used by an enterprise [BD94]. In our case these artifacts are semantic business process models. A business process repository has firstly to provide standard functionality of a database management system, such as storage of new business process models, update, retrieval or deletion of existing business process models, transaction support and query capabilities.

The modeling of business processes can be a time-consuming task. It may take days or even months for business users to finish modeling a given business process. Therefore, treating the entire modeling activity related to a business process model as a single transaction is impractical. A semantic business process repository has to provide check-in and check-out operations, that support long running interactions, enable disconnected mode of interaction with the semantic business process repository, and are executed as separate short transactions. In this case the business process modeling tool works in a disconnected mode regarding the semantic business process repository. The semantic business process model in the semantic business process repository is locked when the business process modeling tool obtains it (check-out), so that no other users can modify the SBP model in the meantime. After the modeling

[1] http://sourceforge.net/projects/iris-reasoner/

work has been done, the process model is updated in the semantic business process repository and any locks that have been held for the business process model are released (check-in). Furthermore, in a distributed modeling environment several business users may work on the same process model simultaneously. A fine-grained locking of elements in a business process model enables different business users to lock only the part of the business process model, they are working on, thus avoiding producing inconsistent business process models.

A business process model may undergo a series of modifications undertaken by business users. The series of modifications is called the change history of the business process model. In certain industry sectors corporations must record all the change histories of their business process models for government auditing or for some legal requirements. From the modeling perspective it is meaningful to keep the change histories of the business process models, so that business users can simply go back to an old state in the change history. Therefore, a business process repository must keep change history of each business process model. Each change step in the change history can be represented as a new version of the business process model in the business process repository. A version is a snapshot of a business process model at a certain point in its change history [BD94].

Actually, there are also more other general repository functionalities that a business process repository could provide, such as configuration control, notification service, consistency checking, user management, and security [BD94]. We will, however, not go through these functionalities in detail, because they are not specific to business process modeling.

In SBPM, business process models are enriched by annotating business process artifacts with entities from pre-defined ontologies. There are different kinds of ontologies that are relevant to business process modeling [HR07], such as organizational ontology, Semantic Web Service ontology, business functions ontology, resource ontology, and domain ontologies. In addition, the business process models themselves are modeled in process ontologies. The ontological descriptions of business process models provide a machine-readable representation of business process models and enable machine-processable reasoning on the ontological descriptions. Reasoning can be used for query answering that is based not only on business process artifacts explicitly stored but also on implicit business process artifacts. Besides the functional requirements identified above, a semantic business process repository must integrate a reasoner for query processing in order to exploit the benefit of ontological annotations. The integration of a reasoner for query processing is also what differentiates a semantic business process repository from a conventional business process repository.

The semantic business process repository that we present in this paper stores semantic business process models described using ontologies, which are formalized using WSML-Flight [WSML05]. Therefore, we need a reasoner that can perform reasoning on WSML-Flight. In SBPM, querying the process space of an organization includes not only ontologies for business process modeling but also domain ontologies, ontologies for enterprise data, organizational structure, Semantic Web Services, and business functions, among others. The instances of these ontologies build datasets which are persisted in the underlying data storage and cannot be handled in main memory, because of their size. Typically, reasoners have to load all the data into main memory, and in that case they are not suitable for the semantic business process re-

pository. For query answering in our case the reasoner must be integrated with the storage mechanism and support loading only the required datasets for reasoning.

3 Storage, Reasoning and Query Processing

In the context of the semantic business process repository, storage, reasoning and query processing issues are interrelated. The semantic business process repository stores instances of process ontologies. The use of ontologies enables using reasoning technology to derive implicit knowledge when answering queries. Thus, the query engine which accesses the store has to be integrated with the reasoner. A comparison of different options of storage mechanisms, reasoners, and their integration is out of the scope of this paper. We will describe in the following one possible solution that satisfies the requirements defined in the last section: we use the Integrated Rule Inference System (IRIS) which is integrated with a relational database system.

3.1 Integrated Rule Inference System (IRIS)

IRIS is an inference engine, which together with the WSML2Reasoner framework[2], supports query answering for WSML-Core and WSML-Flight. In essence, it is a Datalog engine extended with stratified negation[3]. The system implements different deductive database algorithms and evaluation techniques. IRIS allows different data types to be used in semantic descriptions according the XML Schema specification and offers a number of built-in predicates. Functionality for constructing complex data types using primitive ones is also provided.

The translation from a WSML ontology description to Datalog is conducted using the WSML2Reasoner component. This framework combines various validation, normalization and transformation functionalities which are essential to the translation of WSML ontology descriptions to set of predicates and rules. Further on, rules are translated to expressions of relational algebra and computed using the set of operations of relational algebra (i.e., union, set difference, selection, Cartesian product, projection etc.). The motivation for this translation lies in the fact that the relational model is the underlying mathematical model of data for Datalog and there are a number of database optimization techniques applicable for the relational model. Finally optimized relational expressions serve as an input for computing the meaning of recursive Datalog programs.

The core of the IRIS architecture, see Figure 1, is defined as a layered approach consisting of:

- Knowledgebase API;
- Invocation API;
- Storage API.

[2] WSML2Reasoner framework: http://tools.deri.org/wsml2reasoner/

[3] IRIS is continuously being developed and the support for non-stratified negation and unsafe rules is envisioned in coming releases.

The knowledgebase API is a top API layer encapsulating central abstractions of the underlying system (e.g., rule, query, atom, tuple, fact, program, knowledge base, context etc.). The purpose of this layer is to define the basic concepts of data model used in IRIS as well as to define the functionality for the knowledge base and program manipulation.

The invocation API characterizes a particular evaluation strategy (e.g., bottom-up, top-down or mixture of these two strategies) and evaluation methods for a given strategy which are used with respect to a particular logic program.

IRIS implements the following evaluation methods[4]:

- Naive evaluation;
- Semi-naive evaluation;
- Query-subquery (QSQ) evaluation.

The storage layer defines the basic API for accessing data and relation indexing. A central abstraction in this layer is a relation which contains a set of tuples and serves as an argument in each operation of relation algebra. The implementation of IRIS relation is based on Collection and SortedSet Java interfaces where red-black binary search trees are utilized for indexing.

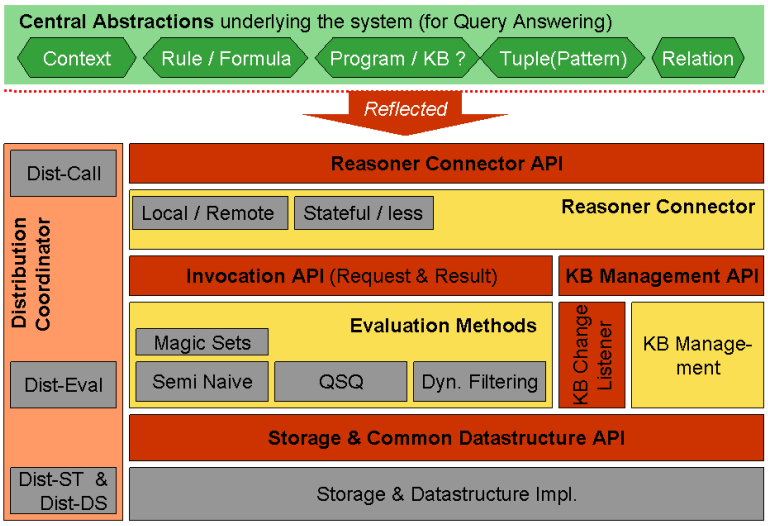

Figure 1: IRIS Architecture

Current inference systems exploit reasoner methods developed rather for small knowledge bases. Such systems either process data in main memory or use a relational database management system to efficiently access and do relational operations on disk persistent relations. Main memory reasoners cannot handle datasets larger than their memory. On the other side, systems based on relational database systems

[4] More evaluation techniques are under development.

feature great performance improvement comparing with main memory systems, but efficient database techniques (e.g., cost-based query planning, caching, buffering) they utilize are suited only for EDB relations and not fully deployable on derived relations.

IRIS is designed to meet requirements for large scale reasoning. Apart from the state-of-the-art deductive methods, the system utilizes database techniques and extends them for implicit knowledge in order to effectively process large datasets. We are building an integrated query optimizer. The estimation of the size and evaluation cost of the intentional predicates will be based on the adaptive sampling method [LN90, RR06], while the extensional data will be estimated using a graph-based synopses of data sets similarly as [SP06]. Further on, for large scale reasoning (i.e., during the derivation of large relations which exceeds main memory), run time memory overflow may occur. Therefore in IRIS we are developing novel techniques for a selective pushing of currently processed tuples to disk. Such techniques aim to temporarily lessen the burden of main memory, and hence to make the entire system capable of handling large relations.

Currently IRIS is a WSML-Flight reasoner. The system is extensively being developed to support reasoning with WSML-Rule (i.e., support for function symbols, unsafe rules and non-stratified negation). Further on, IRIS will tightly integrate a permanent storage system designed for distributed scalable reasoning. One of our major objectives is the implementation of Rule Interchange Format (RIF)[5] in IRIS. Implementing RIF, IRIS will be capable of handling rules from diverse rule systems and will make WSML rule sets interchangeable with rule sets written in other languages that are also supported by RIF.

Finally, IRIS will implement novel techniques for reasoning with integrating frameworks based on classical first-order logic and non-monotonic logic programming as well as techniques for Description Logics reasoning.

3.2 Integration of IRIS with the Semantic Business Process Repository

The semantic business process repository uses a relational database system as the storage mechanism. Relational database systems are widely used both in industry and in research. When using relational database systems, there is no need to re-implement the functionalities such as transaction processing, concurrency control, access control, logging, recovery etc. As relational database systems are widespread for storing data in an organization, using them allows integrating with other enterprise data in a more seamless way.

The needed ontologies, which are formalized in WSML-Flight, are used to generate corresponding relational database schemas. A schema generation tool gets a WSML ontology definition as input and generates the database schema for a particular relational database system (e.g. PostgreSQL[6]) described in SQL Data-Definition Language (DDL).

[5] Rule Interchange Format-W3C Working Group: http://www.w3.org/2005/rules/
[6] http://www.postgresql.org/

When processing queries, the semantic business process repository forwards the query, which is formulated as a WSML logical expression to the IRIS Knowledgebase API. IRIS translates the WSML logical expression to relational algebra statements, from which concrete SQL statements for a particular relational database system are generated.

4 Overall Architecture

In this section, we present the overall architecture of the semantic business process repository. The semantic business process repository has been designed in a layered architecture style consisting of

- Semantic Business Process Repository API;
- Service Layer;
- Persistence Layer.

These three layers are implemented on top of a relational database system. The database schemas are generated from the used WSML ontologies (section 3.2).

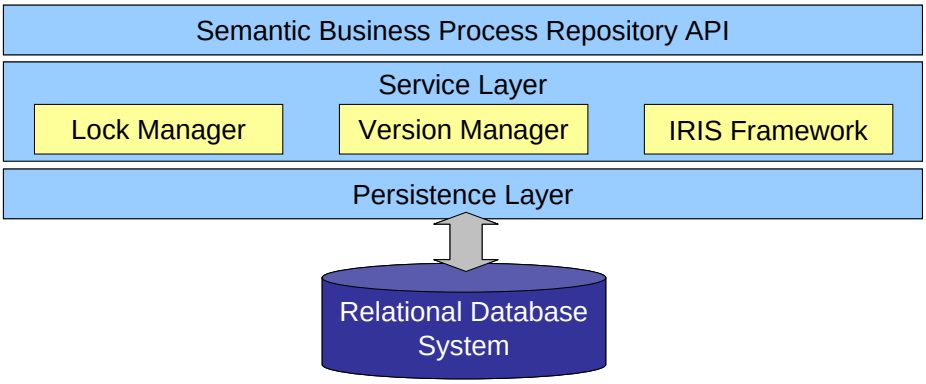

Figure 2: Semantic Business Process Repository Architecture

Semantic Business Process Repository API

The Semantic Business Process Repository API provides programmatic access to the semantic business process repository. It includes an API realizing the CRUD pattern, which represents the four basic functions of persistent storage, namely create, retrieve, update and delete. Besides the CRUD API, the Semantic Business Process Repository API also provides check-in and check-out operations for long-running process modeling. The Query API rounds off the Semantic Business Process Repository API by providing programmatic access to the IRIS Framework for query answering.

Service Layer

The Service Layer implements the Semantic Business Process Repository API and processing logic of the semantic business process repository. The Service Layer contains three modules: Lock Manger, Version Manager and the IRIS Framework. The Lock Manager takes charge of requests on locking and unlocking of the process models in the semantic business process repository. A locking request can only be granted when the process model is not yet locked. The Version Manager takes care of the management of the change histories of process models. To record the change history every new process model or changed process model is stored as a new version in the semantic business process repository. IRIS Framework takes the responsibility for the query processing (section 3.2).

Persistence Layer

The Persistence Layer manages the data access to the underlying relational database system and provides an abstraction for data access operations. It provides persistent solutions for persistent objects by adopting Object Relational Mapping (ORM) middleware such as Hibernate [HIBER] and Data Access Object (DAO[7]) pattern.

5 Summary

In this paper we have presented a semantic business process repository for storage and querying of semantic business process models in SBPM. We have described the functionalities that a semantic business process repository has to provide, namely CRUD API, locking, versioning, and querying using reasoning technology.

In contrast to a conventional business process repository, a semantic business process repository stores instances of process models which are based on ontologies. To exploit the ontological representation, a reasoner has to be used for query processing. Typically, reasoners assume that the whole data is loaded into the main memory, which is not feasible in our case because huge datasets from many different enterprise-related ontologies are needed. As a possible solution we have presented IRIS integrated with a relational database system and described the overall architecture of the repository. We are in the process of integrating IRIS with the relational database system and implementing the semantic business process repository.

Acknowledgements

We would like to thank Marin Dimitrov, Graham Hench, Monika Kaczmarek, Dimka Karastoyanova, Mihail Konstantinov, Tammo van Lessen, Jörg Nitzsche, Jussi Vanhatalo, Karol Wieloch, Paweł Żebrowski and all other colleagues from the SUPER

[7] http://java.sun.com/blueprints/corej2eepatterns/Patterns/DataAccessObject.html

project for valuable discussions. This work has in part been funded through the European Union's 6th Framework Program, within Information Society Technologies (IST) priority under the SUPER project (FP6-026850, http://www.ip-super.org).

References[8]

[BD94] Bernstein, Philip A.; Dayal, Umeshwar: An Overview of Repository Technology. In VLDB 1994.

[HIBER] Hibernate Reference Documentation. Version: 3.2.0.ga

[HLD+05] Hepp, Martin; Leymann, Frank; Domingue, John; Wahler, Alexander; Fensel, Dieter: Semantic Business Process Management: A Vision Towards Using Semantic Web Services for Business Process Management. Proceedings of the IEEE ICEBE 2005, October 18-20, Beijing, China, pp. 535-540

[HR07] Hepp, Martin; Roman, Dumitru: An Ontology Framework for Semantic Business Process Management, Proceedings of Wirtschaftsinformatik 2007, February 28 - March 2, 2007, Karlsruhe (forthcoming).

[LN90] Lipton, Richard and Naughton, Jeffrey. Query size estimation by adaptive sampling (extended abstract). In PODS '90: Proceedings of the ninth ACM SIGACTSIGMOD-SIGART symposium on Principles of database systems, pages 40–46, New York, NY, USA, 1990. ACM Press.

[RR06] Ruckhaus, Edna and Ruiz, Eduardo. Query evaluation and optimization in the semantic web. In Proceedings of the ICLP'06 Workshop on Applications of Logic Programming in the Semantic Web and Semantic Web Services (ALPSWS2006), Washington, USA, August 16 2006.

[SF03] Smith, Howard; Fingar, Peter: Business Process Management. The Third Wave. Meghan-Kiffer,US 2003.

[SP06] Joshua Spiegel and Neoklis Polyzotis. Graph-based synopses for relational selectivity estimation. In SIGMOD '06: Proceedings of the 2006 ACM SIGMOD international conference on Management of data, pages 205–216, New York, NY, USA, 2006. ACM Press.

[SUPER] The European Integrated Project – Semantics Utilised for Process Management within and between Enterprises.
http://www.ip-super.org/

[WSML05] Bruijn, Jos de; Lausen, Holger; Krummenacher, Reto; Polleres, Axel; Predoiu, Livia; Kifer, Michael; Fensel, Dieter: The Web Service Modeling Language WSML. 5 October 2005.
http://www.w3.org/TR/rdf-schema/

[8] All hyperlinks used in this paper were followed on April 10, 2007.