2009

Enabling Distributed Model Management Using Semantic Web Technologies

Amit Deokar
Dakota State University

Omar F. El-Gayar
Dakota State University

Recommended Citation

Deokar, A. V., & El-Gayar, O. F. (2009, January). Enabling distributed model management using semantic web technologies. In 2009 42nd Hawaii International Conference on System Sciences (pp. 1-9). IEEE.

Enabling Distributed Model Management Using Semantic Web Technologies

Amit V. Deokar
Dakota State University
Madison, SD 57042
amit.deokar@dsu.edu

Omar F. El-Gayar
Dakota State University
Madison, SD 57042
omar.el-gayar@dsu.edu

Abstract

In today's increasingly dynamic and competitive business environment, organizations strive to leverage their information resources to gain and sustain competitive advantage. Decision models as knowledge objects encapsulating problem situations, as well as means for leveraging information stored in corporate data warehouses, have positioned such models as an organizational resource that need to be managed, shared and re-used. Recent developments in distributed information technologies and the increasing reliance on such technologies by organization requires that model management accommodates today's distribute infrastructure landscape.

In this paper, we leverage recent developments in semantic web technologies to enable model management functions in a distributed and heterogeneous environment. The proposed architecture leverages OWL to represent models at various levels of abstraction as well as pertinent problem domain, OWL-S to semantically annotate decision models represented as web services, and SWRL to facilitate model querying. Model selection and composition are used to illustrate the applicability of the architecture to model management functions in a distributed setting.

1. Introduction

Over the past decade, organizations have become increasingly focused on managing core competencies in order to gain better competitive advantage, leading to outsourcing of other related competencies. These globalization trends have created complex and dynamic work environments with exacerbated information management challenges. Organizational knowledge assets such as decision models and data need to be managed in a highly effective and efficient manner to predict key outcomes and performance indicators. Moreover, in today's digital world, organizations need to meet the additional challenges of conducting business electronically, while productively managing both inter and intra-organizational resources. Together, the demands of operating in distributed work environments, virtually collaborating with employees and partner organizations, have underscored the need for research on distributed

decision support technologies. These technologies play an important role in enabling the decision-makers by allowing them to focus on making decisions rather than being heavily involved in gathering data, and conceiving and selecting analytical decision models [1].

Distributed model management systems (DMMS) are a particular class of distributed decision support systems with the focus on managing decision models throughout the modeling lifecycle [2]. Models are essentially codified problem formulations, amenable to problem solving techniques such as linear programming or simple linear regression. Model instances represent specific decision making situations created by instantiating model schemas with appropriate data, and are amenable to computational execution using model solvers to determine model solutions. In certain cases, one or more solutions may exist, while many other cases, there may not be a feasible solution to the decision problem at hand. Examples of such models include demand forecasting in a customer service center, production planning to decide optimal product quantities to manufacture, transportation model to deliver the products to the clients under the constraints of time and cost.

Given that model sharing and reuse is one of the primary goals of model management (MM) activities, it is important to develop a model management framework supporting common underlying semantics for model representation and reasoning. Additionally, developing new or adopting existing standards is a key to widespread adoption of such a framework in industry. From a computational perspective, Web services have emerged over the past decade as a standards-based mechanism to describe, encode, publish, retrieve, and compose computational entities such as decision models [3]. From a model representation perspective, markup languages like Structured Modeling Markup Language (SMML) [4] have been recently proposed to facilitate common structural agreement and representation of models. Most recently, semantic web technologies such as ontologies and associated reasoning tools and techniques have seen a variety of applications in providing enriched semantics to knowledge objects. Decision models as knowledge objects are a prime candidate for application of such techniques.

In this paper, we focus on enabling distributed model management through the use of semantic web

technologies. The proposed architecture emphasizes the applicability of different semantic web techniques for different model management activities.

The remainder of the paper is organized as follows. Sections 2 and 3 provide background on distributed model management and relevant semantic web technologies respectively. Section 4 describes semantic representation of models and services based on ontologies. Section 5 presents the proposed architecture, followed by example application scenarios in Section 6. Finally, Section 7 concludes the articles with a summary discussion.

2. Distributed Model Management

Model management has its roots in management science and operations research domains. Since 1980, research in this area has been ongoing to support different model management functionalities including model representation, model manipulation, model selection, model composition, solution computation, result information display and analysis. A comprehensive review of these functionalities can be found in [2, 5, 6].

As noted earlier, with increased globalization, distributed model management has drawn attention from IS researchers beginning mid-1990s. Some of the approaches leveraging distributed computing advances are noted below. Bhargava et al. [7] proposed a web-based architecture for sharing decision models, prototyped as DecisionNet application. The main idea is that of sharing models by publishing and retrieving them through a centralized registry mechanism, similar to yellow pages, by model providers and consumers. Dolk [8] proposed a data warehouse based approach for model storage. It utilizes the structured modeling approach, proposed by Geoffrion [9] for describing models. Huh and Kim [10, 11] proposed a framework for distributed collaborative model management, emphasizing coordination and propagation of changes in a model base in real-time.

Models as loosely coupled components delivering specific functionality can be conceptualized as a service. Likewise, a service as an entity abstracting underlying logic can be considered as a model. A duality between models and services has been noted by Deokar and El-Gayar [12] with respect to reuse, abstraction, autonomy, loose coupling, statelessness, composability, and discoverability. This analogy and duality between models and services indicates a potential for significant synergistic development between model management and service-oriented technologies [13], as noted by some of the following recent advances. Iyer et al. [14] recently proposed a web services architecture for model sharing and reuse of spreadsheet models, while Ezechukwu et al. [15] proposed an architecture for supporting distributed optimization over the Internet. Madhusudan [16] presented a framework for distributed model management

based on web services. The framework utilizes the integrated Service Planning and Execution (ISP&E) [17] for composing web services. Recently, Deokar and El-Gayar [12] presented a semantic web services based architecture for model management systems.

The proposed architecture builds on this past research. The two novel characteristics of this architecture include: (1) truly distributed nature of the architecture, where models as well as different model management functionalities are provided as services, (2) enhancing model and service descriptions with semantic web technologies. The former characteristic relieves the decision maker of performing computationally expensive operations by merely invoking them through a thin client interface. The later characteristic allows reasoning about models in an intelligent manner to support the goals of model sharing and reuse.

3. Semantic Web Technologies

3.1 Knowledge representation and ontologies

Decision models are knowledge objects that capture valuable organizational know-how at operational, tactical, and strategic levels. In order to create computer supported environments such as intelligent decision support systems that encapsulate these models to solve business problems, it is imperative that their representation schemes promote computational reasoning capabilities. In this respect, the notion of Knowledge Representation (KR) is relevant in the context of model representation. KR has a long historical background in Artificial Intelligence (AI) research, where the focus has been on structuring and encoding knowledge in different forms, used in conjunction with inference procedures, for the development of intelligent systems [18]. According to Davis, Shrobe and Szolovits [19], KR plays multiple roles, in that it is a surrogate, a set of ontological commitments, a fragmented theory of intelligent reasoning, a medium for efficient computation, and a medium of human expression.

Ontologies provide a conceptualization mechanism or a vocabulary to represent knowledge in a given domain, and are sometimes referred to as "content theories" [20]. The concept of ontologies has been widely studied and definitions from different perspectives abound. More recently, the notion of ontologies has been referred by the Semantic Web community members as "a set of knowledge terms, including the vocabulary, the semantic interconnections, and some simple rules of inference and logic for some particular topic" [21]. Another relevant definition provided by Studer, Benjamins and Fensel [22] suggests ontologies as "a formal, explicit specification of a shared conceptualization." Researchers have also proposed various categorizations for ontologies. For instance, Guarino [23] distinguishes between top-level

2

ontologies, domain ontologies, task ontologies, and application ontologies.

It is important to note that the major goal of ontologies is not merely to serve as taxonomies or vocabularies, but that of knowledge sharing and reuse by applications and systems. Neches, Fikes, Finin, Gruber, Senator and Swartout [24] point out many modes of knowledge sharing and reuse enabled by ontologies including exchange of techniques, inclusion of source specifications (at design-time), run-time invocation of external modules or services, and interoperability between systems through communication. These modes of sharing and reuse also apply in the context of model management.

Ontological engineering has grown as a subarea within knowledge engineering that concerns with ontology development and use throughout the ontology life cycle – design, implementation, validation, deployment, maintenance, mapping, sharing, and reuse [25].

Ontology representation languages play a key role in ontological engineering by providing a means to build ontologies based on specific KR paradigms to formally represent different knowledge modeling components (such as concepts, and roles). Earlier ontology representation languages such as KIF [26] and Ontolingua [27] were based on KR techniques such as first-order logic and frame-based representation.

3.2 Relevant semantic Web standards

Recently, XML-based ontology representation languages, also called as ontology markup languages, have emerged to support ontology representation in the context of the Semantic Web [28]. Resource Description Framework (RDF) is a W3C (World Wide Web Consortium) recommendation developed for describing Web resources with metadata and incorporates a data model based on the semantic-network KR paradigm [29]. RDF Schema (RDFS) is an extension of RDF with frame-based primitives for defining the relationships between properties and resources, and is also a W3C Recommendation [29]. Ontology Inference Layer (OIL), based on the Description Logics (DL) KR paradigm [30], is an extension of RDF/RDFS adding more frame-based representation primitives and eluding the RDF reification mechanism [31]. DAML+OIL, also based on DL KR paradigm [30], is an evolution of the earlier DARPA Agent Markup Language (DAML) attempting to combine the expressiveness of DAML and OIL by providing DL extensions of RDF/RDFS directly [32].

Web Ontology Language (OWL), derived from the DAML+OIL language, is a W3C recommendation and is the current standard ontology markup language for the Semantic Web [29]. It is extremely rich for describing relationships among class, properties, and individuals. Its vocabulary includes support for property type restrictions, equality, property characteristics, class intersection, and restricted cardinality. Additionally, OWL is not a closed

language; it is instead a combination of three sublanguages with increasing expressiveness, namely OWL-Lite, OWL-DL, and OWL-Full, to support varying needs of knowledge engineers. Recently, Semantic Web Rule Language (SWRL) [29] has been proposed by W3C as a rule language that can be used to write rules in terms of OWL concepts and can reason about OWL individuals. With powerful features like built-ins, which are user-defined predicates, a number of libraries can be custom built (in addition to the core built-ins) for various tasks such as unit or currency conversion, taxonomy queries, and so forth. One such built-in library is the Semantic Query-Enhanced Web Rule Language (SQWRL) in order to support querying of OWL ontologies. SQWRL can be used to build retrieval specifications for knowledge extraction from OWL ontologies.

Web services as executable versions of models are a key component of our architecture as discussed in Section 5. Currently, web services are described procedurally using the Web Services Description Language (WSDL), which lack semantic descriptions of web services. Several approaches have been proposed to adding semantics to web service descriptions. Submissions [33, 34] to the W3C consortium exemplify these approaches: OWL Web Ontology Language for Services (OWL-S), Web Services Modeling Ontology (WSMO), Semantic Web Services Framework (SWSF), and Web Service Semantics (WSDL-S). We use OWL-S, as discussed in Section 4.2, given its process-oriented nature amenable to functionalities such as model composition.

4. Semantic Model / Service Representation

4.1 Semantic representation of models

Ontologies can be used to develop semantically rich models that can support intelligent reasoning and querying based on not only syntactic information, but also semantic information. These reasoning capabilities provide the necessary technological support needed to discover, interpret, compose, and execute models. Moreover, the use of ontologies facilitates the capture of model semantics that is independent of a particular tool or application.

As shown in Figure 1, the different abstraction levels for model representation, as discussed below serve as one dimension. Along an orthogonal dimension are the different domains. A number of domains may also be relevant in the context of a particular problem domain. For example, a transportation model may be primarily formulated for the supply chain domain. The supply chain domain ontology may consist of key terms such as supplier, demand and customer. Other domain ontologies can provide additional semantics to the model structure. For example, currency ontology as mentioned in the discussion above may be used to provide semantics to

3

cost variables. Another units ontology may provide semantics associated with units such as tons. Thus, in a nutshell, the problem domain ontology along with other auxiliary domain ontologies together forms a library of relevant ontologies to provide semantics to the models.
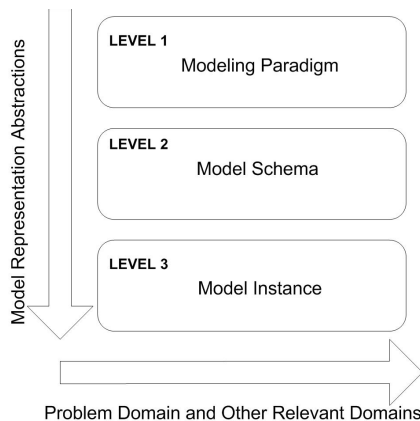

Figure 1. Model Representation Abstractions

Model representation can be viewed at three levels of abstraction. Level 1 indicates the highest level of abstraction, where the goal of representation is to denote a particular modeling paradigm in terms of its fundamental constructs and relationships among them. The overall notion is similar to meta-modeling that gives information about the feasible structure of a particular model schema or instance. We have focused on using Structured Modeling (SM) [9] as the model representation paradigm. Thus, model representation at level 1 provides the concepts and relationships that be used to represent both the model schemas and model instances, from a structural viewpoint. We have built two ontologies using OWL to capture the concepts and relationships in SM for representing model schemas and model instances respectively (refer Figure 2).

The next lower level of abstraction is level 2, where the goal is to represent a particular model schema, independent of its data, such that various sets of data values may be used to instantiate this model at the lowest level of abstraction, i.e. at level 3. For example, an optimization model for a transportation problem in the supply chain domain may be represented in a data-independent manner at level 2. Level 2 essentially is an instantiation of level 1 model schema structure ontology in the context of the domain knowledge. This implies linking it with classes in the problem domain ontologies such as supply chain ontology or other relevant ontologies such as currency (for cost variables). Thus level 2 captures semantics, both in terms of the model structure and problem domain semantics. This provides meaningful information that can be queried and extracted for various model management functions such as model selection and model composition, as discussed in Section 6.
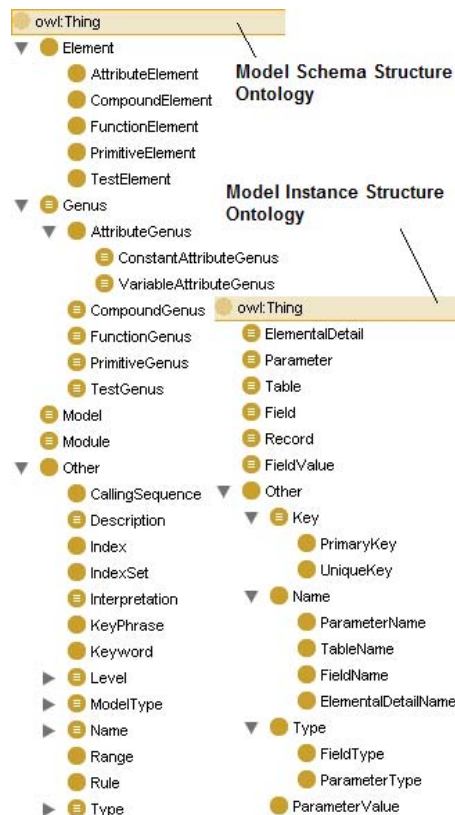

Figure 2. Model Representation (Level 1)

The next lower level of abstraction is level 2, where the goal is to represent a particular model schema, independent of its data, such that various sets of data values may be used to instantiate this model at the lowest level of abstraction, i.e. at level 3. For example, an optimization model for a transportation problem in the supply chain domain may be represented in a data-independent manner at level 2. Level 2 essentially is an instantiation of level 1 model schema structure ontology in the context of the domain knowledge. This implies linking it with classes in the problem domain ontologies such as supply chain ontology or other relevant ontologies such as currency (for cost variables). Thus level 2 captures semantics, both in terms of the model structure and problem domain semantics. This provides meaningful information that can be queried and extracted for various model management functions such as model selection and model composition, as discussed in Section 6.

Similarly, level 3 indicates the lowest level of model abstraction, where the goal is to represent a particular model instance. It is essentially an instantiation of level 1 model instance structure ontology in the context of a particular model schema and a data set. For example, an optimization model for a transportation problem in the supply chain domain may be instantiated with a particular data set where the parameter values needed to solve the

problem are clearly provided. Model solvers can then use this model instance to provide results for the transformation problem for the given data set.

4.2. Semantic representation of services

We use OWL-S [33] for providing semantics to models, encapsulated as executable web services in the proposed architecture. OWL-S is an OWL-based Web Service Ontology language, whose objective is to provide a vocabulary for encoding rich semantic web service descriptions. Service descriptions may be provided using OWL-S that mainly consists of three interrelated sub-ontologies for the top-level concept Service, namely ServiceProfile, ServiceModel, and ServiceGrounding.

The service model (relevant in the context of MM) provides essentially a process model specification to describe 'how the service works' (in other words, how the client may interact with the service), in the form of Inputs, Outputs, Preconditions, and Results (typically called IOPR model), which may be used for service seeking, composing service descriptions, coordinating and monitoring of service executions. The Result concept allows a mechanism to specify several mutually exclusive results with corresponding outputs and possible effects (using the inCondition, withOutput, and hasEffect properties). A process can be one of the three types: atomic (single interaction), composite (combination of processes with some workflow control structure), or simple (service abstraction).

The IOPR model can be specified in any appropriate representation language. We have chosen to use SWRL as the representation of choice, given the amenability of this rule-based representation for sequential composition of services [35]. The Preconditions, inConditions, and Effects are represented using logical formulae in SWRL, expressed over the Input and Output, which are essentially SWRL variables. The Parameter class is an intermediate class between these concepts, and is associated with a hasParameterType property. While the parameter type may be either a class or an XML schema datatype, in order to provide semantically richer descriptions of models (e.g., a GenusName concept is much richer than, say, a string datatype), we take the stance of declaring Inputs and Outputs as concepts from the problem domain ontology or any other relevant ontology. Thus, the conditions expressed in terms of inputs and outputs will also be expressed will also be in terms of the relevant domain ontologies, and as a result have appropriate representation level (as discussed in Section 4.1).

In order to ensure decidability, we restrict the SWRL rules to be DL-Safe [36], i.e. allowing the SWRL rule to bind only to known individuals in an ontology. This check is performed by the rule engine, thus assuring that the services selected based on this reasoning can indeed be executed. Certain other guidelines required for ease of reasoning are mentioned in [35].

5. Proposed Model Management Architecture

Figure 3 shows the proposed model management architecture based on semantic web technologies. At the core of the architecture is the semantic representation of models/services. Executable models are denoted as model web services, which are described semantically using OWL-S and SWRL, as discussed in Section 4.2. Additionally, models may also be represented using either markup languages such as SMML [4] or other formats such as GAMS, LINGO or MATLAB, which can be semantically represented in the form of structured model ontology, as discussed in Section 4.1. These semantic representations of models or services use the terminology from relevant domain knowledge ontologies. Representative model preprocessing and execution services are shown on the right side of Figure 3, with which the user may interact through a front end. In turn, these services leverage semantic processing services such as semantic web rule service and ontology reasoning service. UML component symbol is used to identify executable services. The arrows denote service requests, and not the information exchange occurring. Now, we discuss the interactions among various components of the architecture.

In cases where models exist in SMML format, the ontology mapper service [37, 38] may be invoked to convert the XML representation to a semantic representation based on the SM ontology. In cases where models exist in other formats such as GAMS, the model translator service may be invoked to translate the model into an SMML format, which in turn can be mapped to the SM ontology using the ontology mapper service. SMML models may be queried directly using mechanisms such as XQuery, while the ontology representations can may be queried using SQWRL (refer Section 6.1) to provide semantically richer results. In cases where only executable models in the form of web services exist, the modeler may annotate them using the semantic annotation service to conduct further reasoning
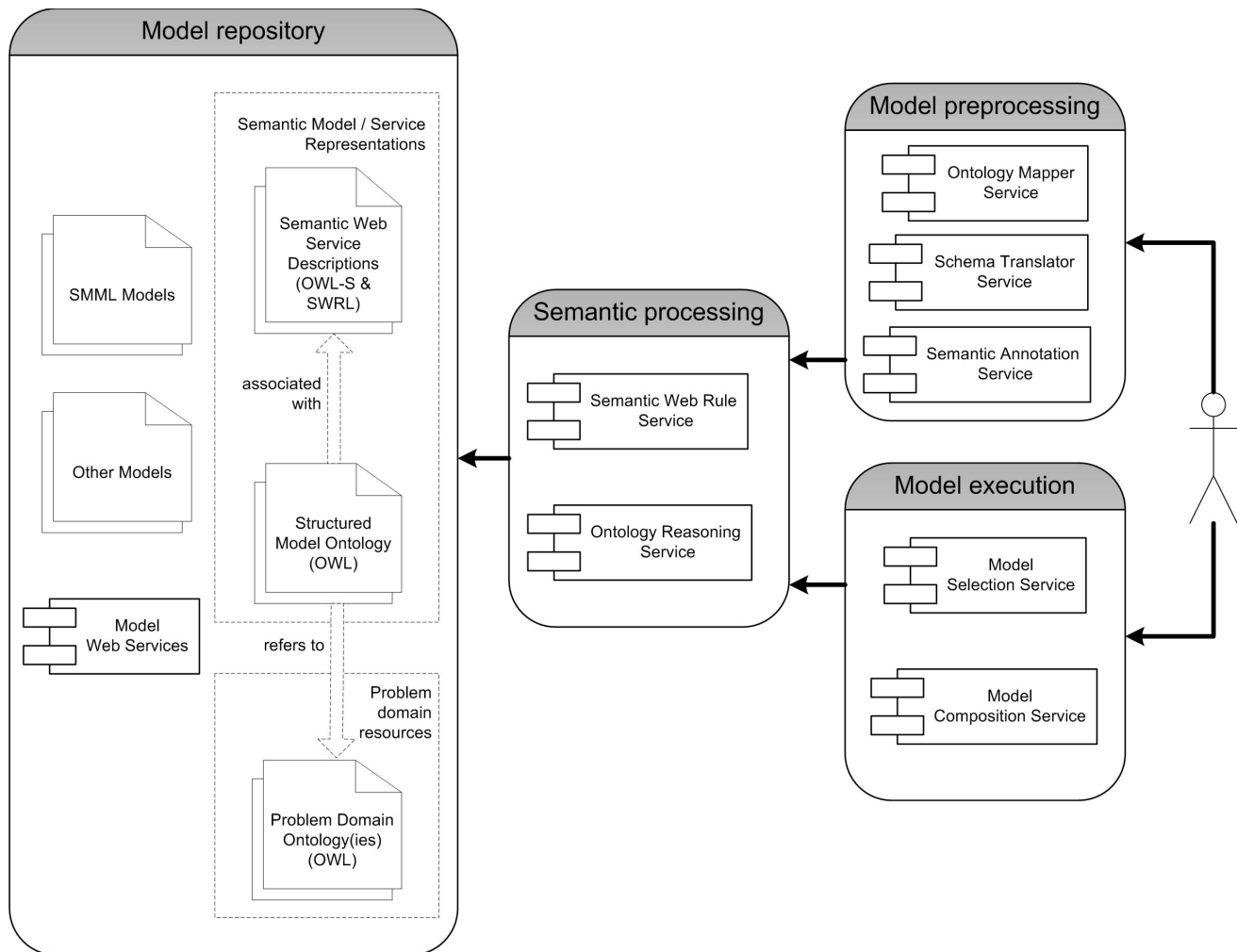
Figure 3. A model management architecture based on semantic web technologies

about the encapsulated model. Since explicit structural information will be missing from their representations, only limited reasoning about such models may be done (e.g., querying based on structural information may not be done). Such selection of relevant models is done by the model selection service using the semantic processing services. The ontology reasoning service reasons about the OWL ontologies to create inferences such as subsumption based on description logic. Ontology reasoners such as Pellet [39] are used for this service. The semantic web rule service is a rule engine (e.g., Bossam [40]) that reasons based on the SWRL rules. The model execution services are discussed as application scenarios in Section 6.

In a nutshell, the proposed architecture leverages semantic representations of models and their execution counterparts, i.e., model web services along with reasoning services to support model management functionalities.

6. Example Application Scenarios

In the discussion below, two types of potential application scenarios are presented to illustrate the ideas discussed earlier. Each of these scenarios have been prototypically tested for feasibility.

6.1. Model selection

Decision makers often need to select an appropriate model for the task at hand based on a number of factors such as the model-task compatibility, data requirements for candidate models, and last but not least execution behavior of candidate models. Here, we draw our attention to model-task compatibility, which can be judged by decision makers based on the results obtained by querying a semantic model base for relevant concepts.

The model ontology repository consists of structured model ontologies (transformed from SMML) and also the corresponding service descriptions of those models. The structured model ontologies in turn refer to particular

domain ontologies. For example, a demand forecasting model in a production environment will refer to terminology using one or more manufacturing domain ontologies. SQWRL queries written in terms of the structured model ontology terms can enable semantic querying of models.

Shown below is a sample query using SQWRL.

```
Genus(?g) ^ hasLevel(?g, ?l) ->
sqwrl:select(?g,?l)
```

This query will return pairs of genera and their structured model levels. The `sqwrl` prefix is used to denote SQWRL operators. Implicit in this query is the information that the genus could belong to any one of the finer genus types (primitive, compound, attribute, function, and test), represented as subclasses of the class genus. Semantic querying of a model base thus allows one to obtain information without knowing the detailed syntactic structure of the model base. In other words, semantic queries support retrieval of both explicit and implicit information based on syntactic and semantic information in a knowledge base. The declarative approach allows user to specify what information is needed without posing the burden of knowing how the information is actually represented. These declarative SQWRL queries can then be reasoned using rule engines such as Jess or Bossom, to find the result of the query. This is supported by the semantic web rule service in the proposed architecture.

6.2. Model composition

Model composition is the problem of generating a sequence of models from a library of available models in response to a particular decision-making situation. Model composition focuses on assembling models together at a functional level [16, 41-47], rather than at a structural level. Only few research proposals attempt to address model composition in distributed settings [7, 48-51].

Using the semantic description of web services in terms of OWL-S and SWRL-based IOPR model (refer Section 4.2) as the knowledge base and a model composition request specified as a SWRL atom, the composer service invokes the semantic web rule service to find if a feasible SWRL rule path exists, and generating all possible such paths. The semantic web rule service is essentially a forward rule chaining rule engine, Bossam [40], which reasons based on the SWRL DL-Safe rules. It can be noted that besides the horn clause based rule base, the inference is also based on the relevant description logic-based domain ontologies. This type of composition is suitable for models in which the process model consists of only atomic processes.

Alternatively, the composition service may be based on Hierarchical Task Network (HTN) planning techniques as discussed in [12]. This composition can be more amenable to OWL-S composite services involving various workflow control constructs [52]. However, in this case, an intermediate task of transforming the SWRL rules in OWL-S IOPE models to SHOP2 planner amenable operators and methods needs to be performed.

7. Discussion and concluding remarks

Analogous to data, this paper acknowledges decision models as an organizational resource that need to be managed, shared and reused. Distributed and heterogeneous information technology infrastructure imposes specific challenges that must be addressed in unique and novel ways. In that regard, the paper proposes an architecture that enables model management function in a distributed setting. The architecture is unique with respect to: (1) It is truly distributed in the sense that model repositories as well as model management functions are distributed. (2) It explicitly leverages semantic web technologies such as OWL, OWL-S, and SWRL to facilitate model management function such as model selection and model composition.

The complexity of the technical landscape and the diversity of modeling paradigms represent significant challenges and opportunities for extending this research. For example, much of the research in ontological engineering comes to bear in the context of model management. Of specific relevance is ontology development and ontology integration for desperate models. Other areas include the role of SWRL in ontology query as well as composition. The latter may include comparing and contrasting SWRL to other composition approaches such as HTN planning techniques as well as exploring areas of complementarities. Moreover, the suitability of the proposed approach to various types of models is something to consider. While structured modeling provided an underlying conceptual foundation for representing the details of mathematical models, structured modeling has its limitations and may not be suitable for all types of models. Last but not least, the complexity of developing ontology-based representation of models will need to be further explored if this research is to transition to practice.

8. References

[1] E. Turban, J. E. Aronson, T.-P. Liang, and R. Sharda, *Decision Support and Business Intelligence Systems*, Eighth ed.: Pearson Prentice Hall, 2007.

[2] R. Krishnan and K. Chari, "Model management: Survey, future research directions and a bibliography," *Interactive Transactions of OR/MS,* vol. 3, 2000.

[3] A. Arsanjani, L.-J. Zhang, M. Ellis, A. Allan, and K. Channabasavaiah, "S3: A service-oriented reference architecture," *IT Pro,* vol. May/June, pp. 10-17, 2007.

[4] O. F. El-Gayar and K. Tandekar, "An XML-based schema definition for model sharing and reuse in a distributed

environment," *Decision Support Systems,* vol. 43, pp. 791-808, 2007.

[5]   R. W. Blanning, "Model management systems : An overview," *Decision Support Systems,* vol. 9, pp. 9-18, 1993.

[6]   A.-M. Chang, C. W. Holsapple, and A. B. Whinston, "Model management issues and directions," *Decision Support Systems,* vol. 9, pp. 19-37, 1993.

[7]   H. K. Bhargava, R. Krishnan, and R. Muller, "Decision support on demand: Emerging electronic markets for decision technologies," *Decision Support Systems,* vol. 19, pp. 193-214, 1997.

[8]   D. R. Dolk, "Integrated model management in the data warehouse era," *European Journal of Operational Research,* vol. 122, pp. 199-218, 2000.

[9]   A. M. Geoffrion, "An introduction to structural modeling," *Management Science,* vol. 33, pp. 547-588, May 1987.

[10] S. Y. Huh, Q. B. Chung, and H. M. Kim, "Collaborative model management in departmental computing," *INFOR* vol. 38, pp. 373-389, Nov 2000.

[11] S. Y. Huh and H. M. Kim, "A real-time synchronization mechanism for collaborative model management," *Decision Support Systems,* vol. 37, pp. 315-330, Jun 2004.

[12] A. V. Deokar and O. F. El-Gayar, "A semantic web services-based architecture for model management," in *Proceedings of the 41st Annual Hawaii International Conference on System Sciences (HICSS-41 '08)* Waikoloa, Big Island, HI: IEEE Computer Society, 2008.

[13] D. F. Ferguson and M. L. Stockton, "Service-oriented architecture: Programming model and product architecture," *IBM Systems Journal,* vol. 44, pp. 753-780, 2005.

[14] B. Iyer, G. Shankaranarayanan, and M. L. Lenard, "Model management decision environment: a Web service prototype for spreadsheet models," *Decision Support Systems,* vol. 40, pp. 283-304, 2005.

[15] O. C. Ezechukwu and I. Maros, "OOF: Open Optimization Framework," Department of Computing, Imperial College London, Departmental Technical Report 2003/7, April 2003.

[16] T. Madhusudan, "A web services framework for distributed model management," *Information Systems Frontiers,* vol. 9, pp. 9-27, Mar 2007.

[17] T. Madhusudan and N. Uttamsingh, "A declarative approach to composing web services in dynamic environments," *Decision Support Systems,* vol. 41, pp. 325-357, January 2006.

[18] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, Second ed.: Prentice Hall, 2002.

[19] R. Davis, H. Shrobe, and P. Szolovits, "What is knowledge representation?," *AI Magazine,* vol. 14, pp. 17-33, 1993.

[20] B. Chandrasekaran, J. R. Josephson, and V. R. Benjamins, "What are ontologies and why do we need them?," *IEEE Intelligent Systems,* vol. 14, pp. 20-26, January-February 1999.

[21] J. Hendler, "Agents and the semantic Web," *IEEE Intelligent Systems,* vol. 16, pp. 30-37, March 2001.

[22] R. Studer, V. R. Benjamins, and D. Fensel, "Knowledge engineering: Principles and methods," *Data and Knowledge Engineering,* vol. 25 pp. 161-197, March 1998.

[23] N. Guarino, "Formal ontology in information systems," in *Proceedings of Formal Ontology in Information Systems (FOIS '98)*, N. Guarino, Ed. Trento, Italy: IOS Press, 1998, pp. 3-15.

[24] R. Neches, R. E. Fikes, T. Finin, T. R. Gruber, T. Senator, and W. R. Swartout, "Enabling technology for knowledge sharing," *AI Magazine,* vol. 12, pp. 36-56, 1991.

[25] A. Gomez-Perez, O. Corcho, and M. Fernandez-Lopez, *Ontological Engineering: With Examples from the Areas of Knowledge Management, e-Commerce and the Semantic Web*: Springer, 2004.

[26] M. R. Genesereth and R. E. Fikes, "Knowledge Interchange Format, Version 3.0 Reference Manual," Computer Science Department, Stanford University, Stanford, CA Logic-92-1, 1992.

[27] T. R. Gruber, "Ontolingua: A mechanism to support portable ontologies," Knowledge Systems Laboratory, Stanford University, Stanford, CA KSL-91-66, March 1992.

[28] J. R. G. Pulido, M. A. G. Ruiz, R. Herrera, E. Cabello, S. Legrand, and D. Elliman, "Ontology languages for the semantic web: A never completely updated review," *Knowledge-Based Systems,* vol. 19, pp. 489-497, 2006.

[29] W3C, "W3C Recommendations: (1) Resource Description Framework (RDF): Concepts and Abstract Synatax, (2) RDF Vocabulary Description Language 1.0: RDF Schema, (3) OWL Web Ontology Language Semantics and Abstract Syntax, (4) A Semantic Web Rule Language (SWRL) Combining OWL and RuleML," http://www.w3.org/2001/sw/, 2004.

[30] F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. Patel-Schneider, "The Description Logic Handbook: Theory, Implementation and Applications," Cambridge, MA: Cambridge University Press, 2003.

[31] D. Fensel, I. Horrocks, F. van Harmelen, S. Decker, M. Erdmann, and M. Klein, "OIL in a nutshell," in *Proceedings of the 12th European Workshop on Knowledge Acquisition, Modeling, and Management (EKAW '00)*, R. Dieng, Ed., 2000, pp. 1-16.

[32] F. van Harmelen, P. Patel-Schneider, and I. Horrocks, "Reference description of the DAML+OIL ontology markup language," http://www.daml.org/2001/03/reference March 2001.

[33] OWL-S Coalition, "OWL-S 1.2 PreRelease," http://www.ai.sri.com/daml/services/owl-s/1.2/, 2006.

[34] W3C, "W3C Members Submissions: (1) OWL Web Ontology Language for Services (OWL-S), (2) Web Service Modeling Ontology (WSMO), (3) Semantic Web Services Framework (SWSF), (4) Web Service Semantics: WSDL-S," http://www.w3.org/2002/ws/swsig/, 2005.

[35] D. Redavid, L. Iannone, T. Payne, and G. Semeraro, "OWL-S Atomic Services Composition with SWRL Rules," in *Foundations of Intelligent Systems*, 2008, pp. 605-611.

[36] B. Motik, U. Sattler, and R. Studer, "Query answering for OWL-DL with rules," *Journal of Web Semantics: Science, Services and Agents on the World Wide Web,* vol. 3, pp. 41-60, 2005.

[37] C. Quix, D. Kensche, and X. Li, "Matching of Ontologies with XML Schemas Using a Generic Metamodel," in *On the Move to Meaningful Internet Systems 2007: CoopIS, DOA, ODBASE, GADA, and IS*, 2007, pp. 1081-1098.

[38] C. Tsinaraki and S. Christodoulakis, "Interoperability of XML Schema Applications with OWL Domain Knowledge

and Semantic Web Tools," in *On the Move to Meaningful Internet Systems 2007: CoopIS, DOA, ODBASE, GADA, and IS*, 2007, pp. 850-869.

[39] B. Parsia and E. Sirin, "Pellet: An OWL DL reasoner," in *Proceedings of the 3rd International Semantic Web Conference (ISWC '04)*, Hiroshima, Japan, 2004.

[40] M. Jang and J. C. Sohn, "Bossam: an extended rule engine for OWL inferencing," in *Proceedings of the RuleML '04 Conference*, 2004.

[41] M. I. Bu-Hulaiga and H. K. Jain, "An interactive plan-based procedure for model integration in DSS," in *Proceedings of the Twenty-First Hawaii International Conference on System Sciences (HICSS '88)*, Kailua-Kona, HI, 1988, pp. 428-434.

[42] K. Chari, "Model composition using filter spaces," *Information Systems Research,* vol. 13, pp. 15-35, Mar 2002.

[43] K. Chari, "Model composition in a distributed environment," *Decision Support Systems,* vol. 35, pp. 399-413, Jun 2003.

[44] V. Dhar and M. Jarke, "On modeling processes," *Decision Support Systems,* vol. 9, pp. 39-49, 1993.

[45] A. Dutta and A. Basu, "An artificial intelligence approach to model management in decision support systems," *IEEE Computer,* vol. 17, pp. 89-97, September 1984.

[46] J. E. Kottemann and D. R. Dolk, "Process-oriented model integration," in *Proceedings of the Twenty-First Hawaii International Conference on System Sciences (HICSS-21 '92)*, Kailua-Kona, HI, 1988, pp. 396-402.

[47] J. E. Kottemann and D. R. Dolk, "Model integration and modeling languages: A process perspective," *Information Systems Research,* vol. 3, pp. 1-16, 1992.

[48] S. Ba, K. R. Lang, and A. B. Whinston, "Enterprise decision support using Intranet technology," *Decision Support Systems,* vol. 20, pp. 99-134, 1997.

[49] M. Holocher, R. Michalski, D. Solte, and F. Vicuna, "MIDA: An open systems architecture for model-oriened integration of data and algorithms," *Decision Support Systems,* vol. 20, pp. 135-147, 1997.

[50] M. A. Jeusfeld and T. X. Bui, "Distributed decision support and organizational connectivity," *Decision Support Systems,* vol. 19, pp. 215-225, 1997.

[51] M. K. Mayer, "Future trends in model management systems: Parallel and distributed extensions," *Decision Support Systems,* vol. 22, pp. 325-335, 1998.

[52] E. Sirin, B. Parsia, D. Wu, J. Hendler, and D. Nau, "HTN planning for web service composition using SHOP2," *Journal of Web Semantics,* vol. 1, pp. 377-396, 2004.

[53] D. R. Dolk, "Next generation model management model evolution as knowledge dynamics," in *Proceedings of the 23rd IFIP TC 7 Conference on System Modelling and Optimization*, Cracow, Poland, 2007.