Inf Syst E-Bus Manage (2013) 11:93–117
DOI 10.1007/s10257-011-0184-2

ORIGINAL ARTICLE

On semantic annotation of decision models

Amit V. Deokar · Omar F. El-Gayar

Received: 16 September 2010 / Revised: 16 May 2011 / Accepted: 19 November 2011 /
Published online: 8 January 2012

Abstract The growth of service sector in recent years has led to renewed research interests in the design and management of service systems. Decision support systems (DSS) play an important role in supporting this endeavor, through management of organizational resources such as models and data, thus forming the "back stage" of service systems. In this article, we identify the requirements for semantically annotating decision models and propose a model representation scheme, termed Semantically Annotated Structure Modeling Markup Language (SA-SMML) that extends Structure Modeling Markup Language (SMML) by incorporating mechanisms for linking semantic models such as ontologies that represent problem domain knowledge concepts. This model representation format is also amenable to a scalable Service-Oriented Architecture (SOA) for managing models in distributed environments. The proposed model representation technique leverages recent advances in the areas of semantic web, and semantic web services. Along with design considerations, we demonstrate the utility of this representation format with an illustrative usage scenarios with a particular emphasis on model discovery and composition in a distributed environment.

Keywords Service science · Model management · Structured modeling · Semantic annotation · Ontologies

A. V. Deokar (✉) · O. F. El-Gayar
College of Business and Information Systems, Dakota State University,
820 N. Washington Ave., East Hall 7, Madison, SD 57042, USA
e-mail: amit.deokar@dsu.edu

O. F. El-Gayar
e-mail: omar.el-gayar@dsu.edu

1 Introduction

Service organizations are faced with numerous information management challenges in creating service innovations in today's increasingly complex and dynamic environment. Vast amounts of data and myriads of models of reality are routinely used to predict key outcomes in service systems. Decision support systems (DSS) play a key role in facilitating decision making through management of data and models. The basic thrust of such applications is to enable decision-makers to focus on making decisions rather than being heavily involved in gathering data, and conceiving and selecting analytical decision models. Consequently, decision and management sciences are among the important reference disciplines for managing service systems (Metters and Marucheck 2007). Efforts from these disciplines are geared towards providing better decision models to enable effective and efficient decision-making (Machuca et al. 2007; Chase and Apte 2007; Roth and Lenor 2003). Embedded in such models are measurable metrics and key performance indicators that can lead to improved service innovation and productivity.

Sharing and reusing these decision models to support co-creation of value in the service value chain, both at the intra-organizational as well as inter-organizational levels, is one of the key challenges facing service enterprises. This is exacerbated by the fact that models use a myriad of languages and task specific representations that include textual descriptions of problem statements, modeling languages, and graphical notation. While some model representations offer distinct advantages such as model-data independence, others have data intertwined with the model structure. Also, several representations (and modeling environments) may be used within the same service organization for addressing the same type of model underlying a particular service. To share and reuse models in such environments, individual translators need to be developed for each pair of model representation schemes. This solution is not scalable, particularly in the context of distributed service settings. Additionally, existing model representations schemes are often paradigm dependent and are not directly amenable to architectures supporting distributed environments. Further, model representation schemes are restricted to encoding structural information about the model, while leaving out the problem domain semantics. Access to this semantic information is crucial in distributed environments to support interoperability of models with each other as well as with the underlying information systems. In effect, a model representation format that captures the structure and semantics of models as well as preserves model-data, model-solver, and model-paradigm independence is needed.

In this article, we propose a model representation format that addresses the above mentioned requirements with a particular focus on semantics. In that regard, we extend the Structured Modeling Markup Language (SMML) to propose Semantically Annotated Structured Modeling Markup Language (SA-SMML) by incorporating mechanisms for linking semantic models such as ontologies that represent problem domain knowledge concepts. Such a representation format is also amenable to a scalable Service-Oriented Architecture (SOA) for managing models in distributed environments. The proposed model representation technique leverages recent advances in the areas of semantic web, and semantic web services.

Following Peffers et al. (2007), the remainder of the paper is organized as follows: Sect. 2 describes the motivation for the work and problems with existing model representational approaches; Sect. 3 defines the objective of a solution in terms of requirements for semantically rich model representations schemes; Sect. 4 presents the design and development of a model representation scheme, namely SA-SMML, and discusses model delivery considerations in the context of a supporting distributed architecture that conceptualizes models and model management functionalities as services, and emphasizes the use of ontologies in leveraging semantically annotated decision models; Sect. 5 presents a demonstrative scenario illustrating the utility of the proposed SA-SMML representation; Sect. 6 evaluates the proposed model representation by comparing its features with the solution requirements from Sect. 3; Finally, Sect. 7 summarizes the main contributions and highlights limitations and venues for future research.

2 Problem identification and motivation

2.1 Motivational scenarios

Recent developments in IT and related technologies have led to new challenges with respect to the handling and processing of large amounts of data for decision making (often in real-time), creating what Tien (2003) refers to as data rich, information poor (DRIP) problems, i.e., rich in basic transaction data, yet poor in processed data such as derivations, recommendations, and patterns which can form the basis of informed decision making. Decision models employed within a decision informatics paradigm can provide a feasible solution to the data rich, information poor DRIP problem noted above (Tien 2003). Decision informatics is comprised of information and decision technologies and is grounded in three disciplines: data analysis, decision modeling, and systems engineering. While data analysis/fusion is concerned with the capture and initial processing of data, decision modeling employs techniques such as optimization and simulation for explicitly supporting decision making, possibly in real time. Glushko and Tabas (2009) also point out the need for bridging the "front stage" with the "back stage" of service systems through systems for managing relevant models and data. The research described in this paper builds upon the notion of decision informatics, particularly from a model management standpoint, in supporting service management.

Decision models can be used in various phases along the service system life cycle (Sage and Armstrong 2000). For example, demand forecasting models (Mukhopadhyay and Samaddar 2007) can be used in *need assessment/requirements and specification*, while workforce and service portfolio optimization (Wright et al. 2006) can be used in the *design and development* of services. Real-time yield management models (Paschalidis and Tsitsiklis 2000) may be used in services such as hotels and airlines. Data Envelopment Analysis (DEA) (Charnes et al. 1978) may be used for *evaluating service* productivity and provide the basis for further service design modifications.

Models can also be viewed as knowledge objects encapsulating an organization's knowledge about a decision problem in a particular domain. The Consortium for Service Innovation (CSI; 2007) advocates a knowledge management strategy emphasizing the value of knowledge for enabling organizations to build an organizational learning culture to improve service levels, operational efficiency, and ultimately customer satisfaction. In this strategy, practices and processes focus on the creation, use, and evolution of knowledge. The modeling life-cycle as described by Krishnan and Chari (2000) represents a rich domain for knowledge management practices as advocated by CSI. Central to the life-cycle is the creation and management of models which encapsulates the explicit knowledge captured through the process and codified in the form of models.

Last but not least, models can also be viewed as services within a SOA. According to Forrester Research report by Heffner (2011), enterprise interest and use of SOA is expanding, and telecommunication, utilities, financial institutions, and insurance companies are primary sectors using SOA the most, with 80% penetration in large enterprises and 60% in small-medium businesses across these sectors. A similar perspective emphasizing the value of service-oriented technology and management is also shared by Demirkan et al. (2008). By viewing models as services within SOA, models can provide the necessary analytics and decision support in real-time to the flexible (re) configuration of business processes and workflows for service management. The following sub-sections elaborate on a couple of scenarios emphasizing model discovery and composition in an organizational setting.

2.1.1 Model discovery

Finding relevant decision making resources to analyze data and solve specific business problems can be time consuming, where a plethora of modeling resources and applications may exist in varying forms within an organization. For instance, consider a forecasting application that predicts new patient admissions at an emergency hospital care. Different types of predictive analytic models such as time series methods and econometric forecasting methods are available and it may be necessary to choose a certain model based on criteria of interest. The semantic annotation of modeling resources would improve expressivity and inferencing capabilities. This would facilitate and improve operations such as model discovery and composition in a manner that is not possible without semantic descriptions. Moreover, machine understanding of semantics and semantic interoperability would present opportunities for models as services to be discovered and integrated with other applications on an as-needed basis. An example use case for this would be: "find me a weighted moving average forecasting model for the emergency care facility that can take number of patient admissions for the past 30 days and model parameters (number of periods, and weights in this case), and generate a prediction for the following week." In this example, an organization may have several semantically annotated predictive analytic models that it has developed over time for its operations. The objective is to select a particular model that can compute weighted moving average prediction and is appropriate for the underlying decision

problem. Other forecasting models in the model repository may use other techniques such as autoregressive moving average (ARMA), which are not based on the weighted moving average principle and do not take past data, number of periods, and weights as inputs and thus should not be included in the resultant model match. Also, different models in the repository may use different domain specific terminology that needs to be reconciled. This is possible only through the inferencing on the semantic models, i.e. ontologies, rather than a simple keyword based search. Accordingly, in this use case, model discovery involves matching the requirements to extract only the relevant models from a larger model repository by leveraging semantics embedded in the model representation.

2.1.2 Model composition

In cases where a single model does not exist to address the decision problem at hand, it may be necessary to chain together a series of models to achieve the desired goal. An example use case for this in the emergency care facility scenario would be: "given a set of decision models, find me a composition of models to compute the required nurse staffing levels at the emergency care facility for the upcoming week so as to minimize the number of nurses required to staff the facility." To be able to determine the desired nurse staffing levels for a week, first the patient demand will need to be forecasted for the upcoming week using a patient admission forecasting model, following which a staff scheduling model like an integer linear programming model will need to be invoked. During this process, the forecasted demand obtained for each day in the following week will need to be used as the desired staffing level for each day in the integer linear programming formulation. In identifying relevant candidates that meet the goals of the composition request, machine-understandable and expressive semantics will play an important role. Different terms (e.g., 'forecast_admission', 'desired _nurse_staff_level') used by various decision models will need to be reconciled through semantic models such as domain ontologies, regardless of the specific model composition technique used in finding candidate model compositions based on model specific inputs, outputs, preconditions, and effects.

2.2 Model representation supporting model management functions

Based on the aforementioned discussion, the significance of the role of decision models in the service value chain and service management is evident. As such, distributed decision support systems in general, and model management systems in particular, are an important part of the IT infrastructure in the back stage of service systems. Model management (MM) (Blanning 1993; Chang et al. 1993) encompasses a variety of functionality including model description, model manipulation, scheduling, execution, and information display (Muhanna 1993b). Much of the model management functionality rests of the ability to represent models at a higher-level of abstraction, i.e., meta-modeling (Muhanna and Pick 1994).

Numerous proposals and languages have been put forth in the literature for model representation. Algebraic modeling languages (e.g., GAMS (Brooke et al. 1988),

AMPL (Fourer et al. 1993), and LINGO (Katz et al. 1980)) use matrix generators or related mechanisms to abstract away low-level computational details and allow the knowledge worker to deal with models represented using symbolic, general, and concise means (Fourer 1983). These languages, however, are often modeling paradigm specific, and do not support meta-level reasoning of models (Krishnan and Chari 2000). Also, model representation requirements support varies from language to language. GAMS (Brooke et al. 1988) supports model-solver independence, while AMPL (Fourer et al. 1993) supports model-data and model-solver independence.

Logic-based approaches such as those proposed by Bhargava and Kimbrough (1993) provides meta-model support by augmenting existing modeling languages by formally representing information about expressions, typically not amenable for representation. Such meta-model information can be used for MM functionalities. The knowledge worker however, now has the additional burden of handling models in different representational formats through the embedded language approach proposed.

Another scheme for model representation has focused on using data-oriented approaches to model representation (Dolk 1988; Lenard 1986; Liang 1985a; Miller and Katz 1986; Stohr and Tanniru 1980). In that direction, relational-based approaches, such as (Blanning 1985; Liang 1985b; Choobineh 1991), try to apply the relational database technology for model representation and management. However, such approaches are particularly limited when it comes to handling rich model semantics and are often paradigm dependent.

Given the variety of models that can be represented by graphs (Bunke 1982; Jones 1990), graph-based modeling systems (GBMS) (Jones 1990; Jones 1991) use attributed graphs for representing models. Graph grammars are used in these systems to represent structural constraints imposed on various types of graphs. Some examples of such graphs are vehicle routing, neural networks, and structured modeling Genus graphs. The applicability of graph-based model representation scheme is theoretically appealing, however graph grammars can be difficult to manipulate, and their practical utility to a wide variety of graph-models needs further research (Jones 1990).

Structured modeling (SM) is a model representation technique proposed by Geoffrion (1987), where mathematical models can be conceived as hierarchically organized, acyclic, attributed graphs (Geoffrion 1992a, b). SM provides a conceptual framework for conceiving, representing, and manipulating a wide variety (paradigms) of models. In SM, model solver issues are neatly separated from representation concerns. For example, in case of mathematical programming models, SM does not specify the objective function and whether its value should be maximized or minimized. These solver issues are to be defined at the model execution time. This provides flexibility for a user, who may decide to use some or all of the constraints from the model. The model representations are independent of the model solution and the way the model is solved. Moreover, the model schema does not contain any actual data and this provides model-data independence. The same model may be invoked using different set of data. The data, in SM terminology known as a model instance, is maintained separately than a model

schema. Accordingly, in SM, a model is defined as a combination of a model schema and one or more model instances. A model schema describes the general structure of a model and may be associated with one or more model instances. Model instances correspond to the data part of the model. Detailed description of SM concepts can be found in (Geoffrion 1987, 1989a, 1992a, b).

With the proliferation of the Internet and distributed computing environments, a number of model representation approaches that are based on the Extensible markup language (XML) have been developed for representing models in different problem domain and for specific modeling paradigms. XML has become the de facto standard for message exchange and interoperability support in distributed heterogeneous environments, given its standardized and non-proprietary nature. In the data mining area, Predictive Model Markup Language (PMML) provides a tool-independent representation for predictive models amenable to algorithmic techniques such as regression, cluster analysis, decision trees, neural networks, and Bayesian analysis (Data Mining Group 2010). In the simulation area, National Institute of Standards and Technology (NIST) has been working on simulation interface standards in the context of manufacturing software applications (McLean et al. 2005). Wang and Lu (2002) propose an XML application to represent discrete event simulation models based on the Discrete Event System Specification (DEVS) approach (Zeigler 1990).

With respect to representing graph models, several XML-based languages are available including GraphML (Brandes et al. 2004), GXL (Holt et al. 2002), and NaGML (Bradley 2005). While the overall purpose of these proposals is similar, they differ with respect to domain areas, and implementation features. GraphML was initiated during the 2000 Graph Drawing Symposium in Williamsburg, Virginia as a mean to represent graphs, while GXL evolved from the software re-engineering community with an emphasis on representing and exchanging software engineering artifacts as graphs. NaGML incorporates problem domain schema within the document by allowing users to specify the name, data type, and restrictions for each node and arc property.

A number of XML-based languages have been proposed in the context of optimization models (Bradley 2003). Optimization Service Instance Language (OSiL) is an XML-based computer language proposed by Fourer et al. (2006) for representing instances of large-scale optimization problems including linear programs, mixed-integer programs, quadratic programs, and very general nonlinear programs. Fourer et al. (2005) propose LPFML as an XML schema for representing linear programming (LP) models. Focused on LP models, LPFML aims to standardize the representation of this class of optimization problems. However, the LPFML representation has a strong model-instance focus, and the model-data are intertwined. Also, it does not use a higher level of abstraction such as sets in representing the model schema and instance. Other optimization related proposals include OptML (Kristjansson 2002) and SNOML (Lopes and Fourer 2005), both of which focus on representing instances of linear and mixed integer programming models at the matrix level as XML files. In the context of a comprehensive framework for optimization, Ezechukwu and Maros (2003) propose an architecture supporting distributed optimization over the Internet.

The architecture uses two XML-based languages, namely, Algebraic Modeling Language (AML) for representing models, and Optimization Reporting Markup Language (ORML) for representing model solutions, and a set of programs to convert the model to a target system for execution and converting model results in a desired format.

A couple of XML-based markup languages have been proposed that are based on the SM formalism, namely OOSML (Object-Oriented Structured Markup Language) proposed by Kim (2001), and SMML proposed by El-Gayar and Tandekar (2007). Given their basis on the SM formalism, these languages potentially realize many of the desirable MM features mentioned earlier. However, OOSML is based on the XML Document Type Definition (DTD), which has become almost obsolete and is incapable of representing complex structures. Also, the DTD used for OOSML does not provide support for representing mathematical equations and explicit indexing. SMML, which is based on XML schemas standard and uses MathML to represent mathematical equations is better in that regard. Bhrammanee and Wuwongse (2008) propose ODDM as a framework for leveraging OWL Declarative Description (ODD) for representing decision models. However, a limitation of the approach is the lack of an underlying theoretical foundation such as SM (1987) for model representation.

With the exception of Kim's (2001) OOSML, and El-Gayar and Tandekar's (2007) SMML, a common problem with the aforementioned XML-based representation approaches is that they do not meet model representation requirements such as model-paradigm independence and model-data independence, possibly due to problem domain-specific focus and lack of conceptual foundation for representing models at higher level of abstraction. Further, none of the proposed approaches give considerable attention to accessibility, interoperability, contextual information requirements mentioned earlier. It is not clear how such XML-based models can be shared and reused with the availability of wide variety of modeling languages and formats. In the same vein, concerns about how models with differing formats would interface with application systems remain unaddressed. Last, but not least, challenges stemming from distributed environments related to varying problem domain context information remain unresolved.

3 Requirements for semantically annotating decision models

Model representation research has evolved over the years based on the changing requirements from analysts and end users. The ability to represent models at a higher level of abstraction rather than low-level input formats was one of the main initial requirements. Over the period, more design requirements have emerged (Muhanna 1994; Muhanna and Pick 1994):

- *Model-paradigm independence*: ability to represent models from different modeling paradigms (Geoffrion 1987)
- *Model-data independence*: ability to use the model with different data sets (Geoffrion 1987; Muhanna and Pick 1994)

- *Model-solver independence*: ability to use the model with different solvers (Geoffrion 1987)
- *Meta-level representation and reasoning*: ability to represent information about models (Muhanna and Pick 1994),
- *Common representation format*: interoperable format amenable to model sharing and reuse (Bose and Sugumaran 2007; Geoffrion 1987).

Driven by advances in supporting communication infrastructure, the need for sharing decision models within and across service enterprises as well as deriving value from the application of such models in information systems interfacing the customers has become more apparent. Additional requirements concerning model representation and management have emerged with respect to their use in distributed settings:

- *Accessibility of distributed decision support resources*: ability to store, search, retrieve, utilize, reuse distributed resources such as models and data (Ezechukwu and Maros 2003)
- *Interoperability*: ability to interoperate with different information systems that can utilize models to solve decision problems
- *Portability*: compatibility across various development environments/systems
- *Vendor independence*: avoiding being locked in by a particular vendor
- *Standards-based approaches*: compatibility with accepted WWW and Internet standards and technologies (e.g., XML, web services, semantic web standards)
- *Extensibility*: amenable to extensions and continuous improvement efforts
- *Incorporate semantic information*: ability to capture problem domain semantics, and leverage it for retrieving relevant resources for various use cases

Moreover, to facilitate distributed model management, web services pose as a viable technology to accomplish the mediation task (Erl 2004; Sahai and Graupner 2005). Web services are based on service-oriented computing principles and provide a standardized way of integrating several application modules. Decision models as well as model management functionalities are decision support resources that need to be leveraged in distributed settings. Conceptually, as a loosely coupled component, each model and model management functionality can be conceived as a computational service or web service. Principles underlying service orientation (Ferguson and Stockton 2005), namely (1) reuse, (2) abstraction, (3) autonomy, (4) loose coupling, (5) statelessness, (6) discoverability, and (7) composability are compatible with the objectives of the use of models and model management functionalities representing back stage IT components in service management. They address key model representation requirements of accessibility and interoperability in that they provide a mechanism for sharing models in distributed environments as well as linking them to other application systems.

While the service-oriented perspective of conceiving models and related modeling resources as computational services is a useful mechanism for enabling distributed sharing of these resources, it also adds complexity in terms of dealing with different terminologies, problem contexts, and so forth in distributed environments. As such, decision models need to be augmented with semantics to

be used effectively in IT applications supporting service systems functioning in distributed settings.

The overall rationale for incorporating semantic information in decision models is twofold. First, semantic information can explicate the link between the decision models and domain concepts, which is often implicit in the minds of the modeler and thus unknown to other knowledge workers and software agents. By articulating this link, a contextual view of the model with respect to the domain knowledge can become apparent and be utilized by applications. Second, semantic information can help support interoperability and reuse of models. Models using different terminology but referring to equivalent or similar domain concepts can be search and retrieved using a common ground approach with the semantic information acting as an intermediary. With a better understanding of the underlying semantics, models may be reused for other decision problems by adapting or by combining with other compatible models.

Based on the overall rationale, the objectives for a model representation solution are to accommodate model semantics while still meeting the aforementioned design requirements. Moreover, in accommodating model semantics, the proposed solution should meet design requirements that would facilitate the seamless utilization of models as services in distributed and heterogeneous environments. These requirements include the following:

- *Machine-understandability*: Semantic information incorporated in decision model schemas and instance data needs to be machine interpretable. Software agents and applications forming the back stage of service systems can then computationally operate over this information.
- *Semantic interoperability*: Decision models may exhibit heterogeneity in variety of ways including differing terminology, specificity of information, and so forth. Semantic interoperability between decision models should be supported to address this heterogeneity.
- *Expressivity*: Ontologies and related semantic models used to annotate decision models should be expressive enough to capture domain knowledge concepts that can facilitate software agents to provide an accurate contextual view, and interpretation.
- *Inferencing support*: Semantic annotations in decision models should be able to be reasoned upon using reasoning and/or rule engines to draw requisite inferences. These inferences can be used by knowledge workers such as service operations analysts themselves for further analysis and interpretation, or may be used by software agents to undertake further action.
- *Leveraging existing efforts and standards*: Semantic information incorporated in decision models should leverage existing advances in multiple areas including model representation schemes, semantic web standards, and software architectures.
- *Ease of application development*: It should be relatively easy to develop applications to support tasks related to semantic annotation, reasoning, querying, and so forth. As such, acceptance of standards and tool availability should be considered in proposals pertaining to semantic annotation of decision models.

4 Design and development for the semantic representation of structured models

Considering the extant model representation approaches and languages in association with the requirements, SMML is particularly attractive to extend with additional features for two main reasons. First reason is that it is based on a strong conceptual foundation, namely SM. From a model management standpoint, SM has many characteristics that are highly desirable (Krishnan and Chari 2000; Geoffrion 1987), Specifically, SM has many features that are highly desirable from a MM perspective (Krishnan and Chari 2000; Geoffrion 1987). Most notable ones are model-data independence, model-paradigm independence with sufficient generality to describe different kinds of models, and independence of model and model solution (Geoffrion 1987). Model formulation (Liang and Konsynski 1993), model integration (Geoffrion 1989b) and model composition (Holocher et al. 1997) tasks have demonstrated using the SM representation approach. SM has been used as the basis for a number of other proposals for model representation including object oriented approaches (Lenard 1993; Muhanna 1993a; Huh 1993; Gagliardi and Spera 1997), relational-based approaches such as (Lenard 1986; Dolk 1988), and graphic-based approaches such as GBMS/SM (Chari and Sen 1998). The second reason is that SMML being an XML language is particularly suited for distributed settings because of desirable characteristics of XML in terms of portability across systems and development platforms, vendor neutrality, extensibility, and standards-based approach. Based on these arguments, this research seeks to use SMML as the basis for the proposed model representation scheme for annotating decision models.

4.1 Abstraction levels for model representation

It is useful to consider the different abstraction levels in model representation, as shown in Fig. 1. Typically, model representations can be conceptualized at three levels of abstraction, namely, level 1—modeling paradigm, level 2—model schema, and level 3—model instance. Level 1 indicates the highest level of abstraction, where a particular modeling paradigm is denoted 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. Structured modeling (Geoffrion 1987) and metagraphs (Basu and Blanning 1994) are examples of modeling paradigms. Considering SMML based on SM as an example, the SMML XML schemas for model schema and model instance capture the vocabulary (e.g., genus, module, model) from the SM paradigm, and represent level 1 model abstraction.

Level 2 indicates the next lower level of abstraction, where a particular model schema is represented, independent of data and parameters, such that various data sets and parameters may be used to instantiate this model at the next level of abstraction, i.e. at level 3. A transportation model schema in SMML that adheres to the SMML model schema (XML schema) represents an example of level 2 model abstraction. Level 3 is the lowest level of model abstraction, where a particular model instance is represented. It can be conceptualized as an instantiation of level 1

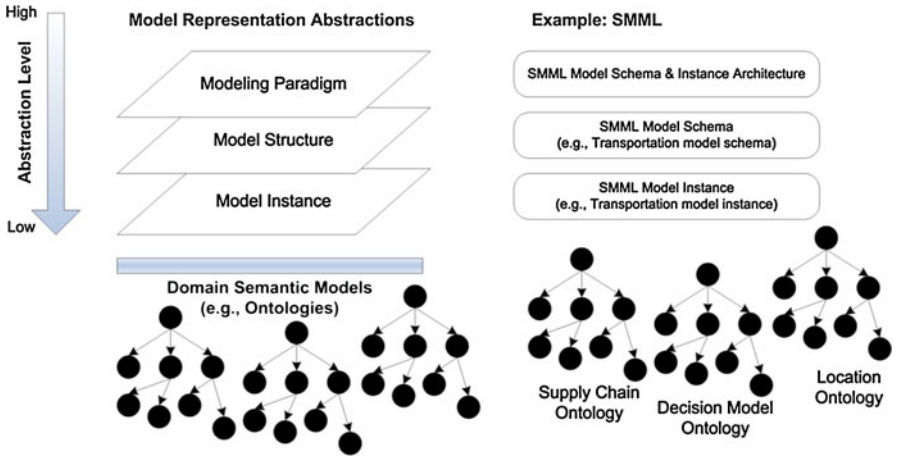

Fig. 1 Model representation abstractions and domain knowledge models

for a particular model structure and a data set. A transportation model instance in SMML that adheres to the SMML model instance (XML schema), and provides parameters and data values for the transportation model structure is an example of level 3 model abstraction.

Orthogonal to the abstraction level are various semantic models representing domain knowledge concepts. Semantic models are domain ontologies or formal knowledge representation structures (e.g., RDFS, OWL, RIF, and so on). Each modeling abstraction may refer to multiple semantic models. Certain semantic models may be common across modeling abstractions. For example, the `FIELD` names used in an SMML model instance should refer to the same domain concepts expressed as `GENUS` names in the corresponding SMML model schema. This link ties the model schema and a model instance together and thus the reference to the same domain concept is important. In the following section, we describe the proposed SA-SMML model representation scheme that describes how models can be annotated with references to semantic concepts.

4.2 Semantic annotation structured modeling markup language (SA-SMML)

The proposed approach for incorporating semantic information in structured models is based on the World Wide Web Consortium's (W3C) standard, namely Semantic Annotations for WSDL and XML Schema (SAWSDL) (Kopecký et al. 2007). Although the development of SAWSDL standard originated from the need to semantically annotate web services, the recommendation is general enough and is a lightweight approach for adding semantics in XML-based languages such as SMML. Using standards-based approach such as SAWSDL as the foundation is certainly advantageous, given the industry momentum behind such efforts in terms of developing associated tools, use cases, and gaining widespread acceptance among practitioners. One of the other significant advantages is the ability to

semantically annotate models represented in varying formats and languages and using an SAWSDL-based approach is useful in addressing this issue.

The extended SMML representation is called Semantically Annotated SMML (SA-SMML). It is a lightweight mechanism in the sense that it provides pointers to semantic models[1] from within SMML, while being agnostic to the specific language used for representing semantic models (e.g., OWL, RDF Schema). Semantic models can vary in form and may simply be agreed upon terminologies in a certain domain developed by service enterprises, or can be extensive formal models such as ontologies using logic-based formalisms like description logics. The decoupling of the semantic models from the decision model itself allows leveraging the development efforts invested in creating ontologies by enterprises, consortiums, and community at large.

Semantic models that are defined outside of SMML are referenced from within SMML. In that regard, three key attributes, namely `semanticReference`, `liftingMapping`, and `loweringMapping` are used. The `semanticReference` attribute points to semantic concepts like classes in domain ontologies, while `loweringMapping` and `liftingMapping` attributes specify data transformations between a decision model's XML structure and the associated semantic model. These attributes have been added to `ModelType`, `GenusType` and `ModuleType` type definitions in SA-SMML model structure schema, as shown in Fig. 2. Figure 3 shows the attribute extension for `ModelType`. Based on this addition to the XML schema, `MODEL`, `GENUS`, and `MODULE` elements in a SA-SMML model schema can provide references to domain concepts recorded externally in semantic models.

Along similar lines, SA-SMML model instance XML schema has been extended from SMML by adding attributes to (a) the `ELEMENTAL_DETAIL` element within the `TableType` type definition, (b) the `FIELD` element within the `RECORD_DESC` element of `TableType` type definition, and (c) the `FIELD` element within the `RECORD` element of `TableType` type definition, in the SA-SMML model instance schema, as shown in Fig. 3. The difference between annotation types (b) and (c) is that the goal of the former is to add semantic annotations to the field names (e.g., `PLANT` denoting `supplier_location` concept in supply chain domain ontology) in the model instance, while the goal of the later is to add semantic annotations to the field values (e.g., `DAL` denoting `DALLAS` in a location taxonomy). An SA-SMML model instance can be annotated with semantic information using these extensions.

4.3 Model delivery considerations

Figure 4 shows a supporting architecture for model sharing and reuse in a distributed setting. The model management platform is shown as the underlying infrastructure supporting service enterprise systems. The model management services provide access and management (e.g. creation, modification, storage, retrieval, deletion of models) to a variety of modeling resources, and act as the glue

[1] The term "semantic models" is used to denote domain knowledge models such as ontologies, while the term "decision models" or simply "models" is used to refer to mathematical models representing decision problem formulations.

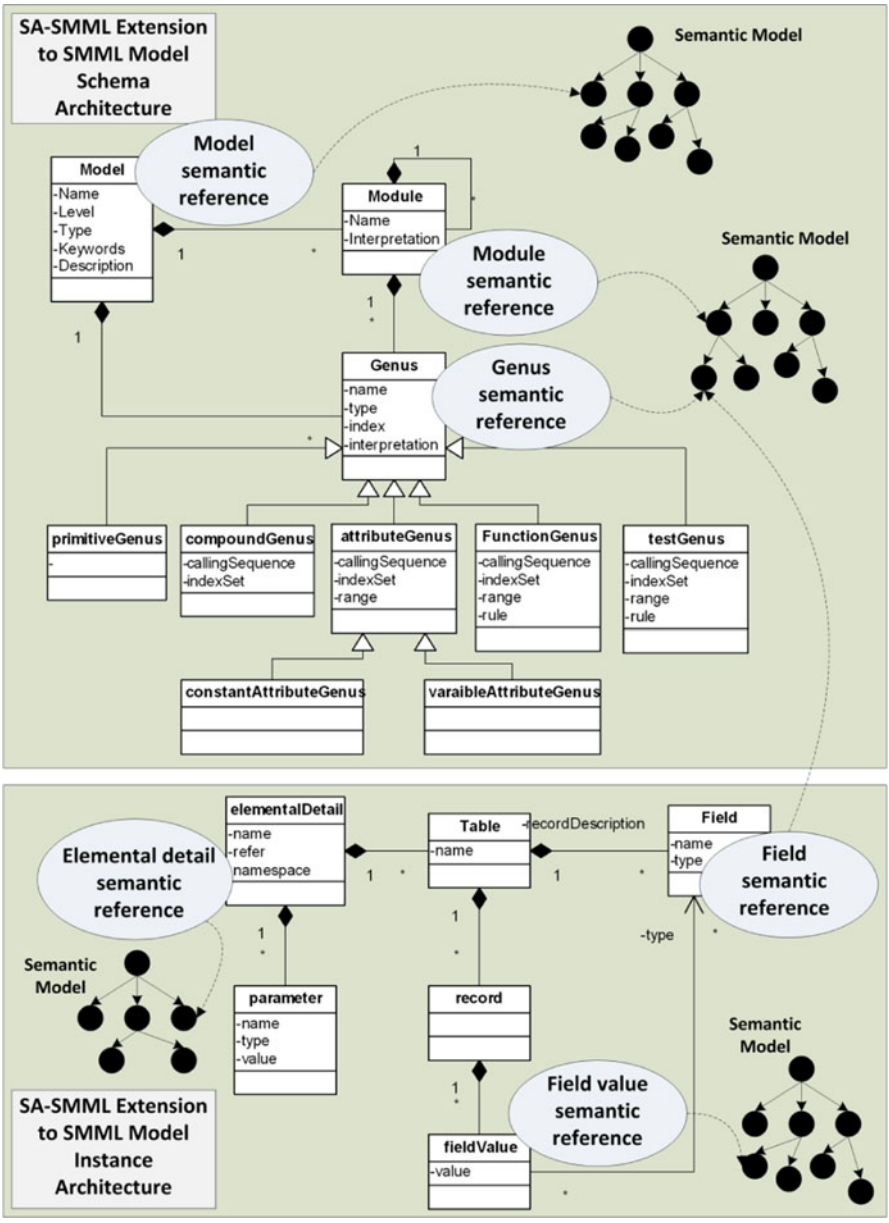

Fig. 2 SA-SMML overview in a UML class diagram

between service enterprise systems and modeling resources. Models as services denote models of different formats encapsulated as services. For executable models, wrapper services are used to encapsulate the functionality of existing modules as web services. For model schemas represented as stand-alone non-executable files

```
<xsd:complexType name="ModelType">
     <xsd:sequence>
          <xsd:choice maxOccurs="unbounded">
               <xsd:element name="MODULE" type="ModuleType"/>
               <xsd:element name="GENUS" type="GenusType"/>
          </xsd:choice>
     </xsd:sequence>
     <xsd:attribute name="name" type="xsd:string"/>
     <xsd:attribute name="semanticReference" type="xsd:anyURI"/>
     <xsd:attribute name="liftingMapping" type="xsd:anyURI"/>
     <xsd:attribute name="loweringMapping" type="xsd:anyURI"/>
     ...
</xsd:complexType>
```

Fig. 3 Semantic extensions in SA-SMML: example of ModelType

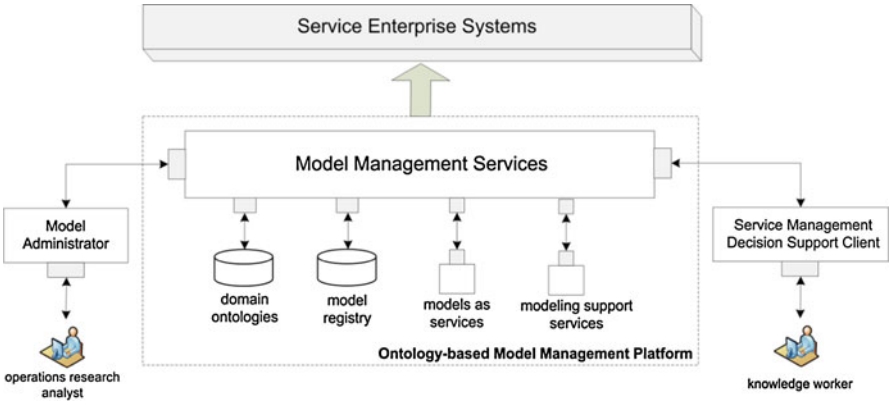

Fig. 4 An architecture for decision-enabled service management

such as GAMS, AMPL, and XML representations of models (e.g., SMML), schema wrapper services provide access to the model parameters and structural details. Model schemas such as GAMS models that can be executed using a solver (e.g., GAMS solver) may alternatively be represented in the form of proxy web services which include operations to execute the models by invoking corresponding solver services.

Services other than models themselves are shown as modeling support services. For specialized model solvers, and development environments and platforms (e.g., AMPL, MATLAB), proxy services are used to expose the functionality afforded by these environments for interfacing purposes. Model management services include publishing services, discovery services, account management services, model translation services, model composition, and model analyzer services.

Models and modeling resources (as services) are registered in a centralized registry accessible to the model management services. A particular characteristic of this architecture is the concept of semantically annotating decision models for

reasoning and conducting different model management operations on them, which is discussed in detail in the following sections. Domain ontologies used for incorporating semantics in models are indicated as a resource in the architecture. The model administrator component provides administrative access to manage models as services. Typically, a modeling expert such as an operations research analyst is likely to be the role member responsible for performing model administration. The client component provides a user interface to access model management functionalities for knowledge workers engaged in service management roles.

Given the existing heterogeneity of decision modeling paradigms and formats, design considerations in terms of how to accommodate such different kinds of models are important. Being an extension of SMML, SA-SMML is sufficiently generic to represent models of different paradigms and can potentially serve as an intermediary format for different modeling resources.

Another aspect of heterogeneity of models has to do with how models with varying formats are accessed and delivered to supporting information systems. As shown in Fig. 5, models fall along a wide spectrum in terms of the specificity of structure and semantics in their representation. Models exhibiting high specificity in terms of model structure and semantics fall toward the right side of the spectrum.

Structured models represented using SA-SMML lie at the extreme right end, and incorporate semantic information in addition to the model structure represented in SMML models. Figure 5 also depicts the approach suitable for semantically annotating models with differing forms and shape. Models in SMML can be seamlessly transformed to SA-SMML models by embedding appropriate semantic links, given their same foundational structure. The SA-SMML model representation format is also the semantic annotation approach for these models. SA-SMML models are then encapsulated as Web services for delivery purposes.

Models that are encoded in other higher level representation formats (e.g., AMPL, GAMS, and LINGO) fall midway along the spectrum. Among these models, some models may be inherently compatible with the SM paradigm (e.g. GAMS optimization models (GAMS Development Corporation 2010)), while some may not (e.g., recursive models (Geoffrion 1987)). SM compatible models may be annotated in a couple of different ways. If such models are translated to SA-SMML models using model translator services, then semantic links can be incorporated

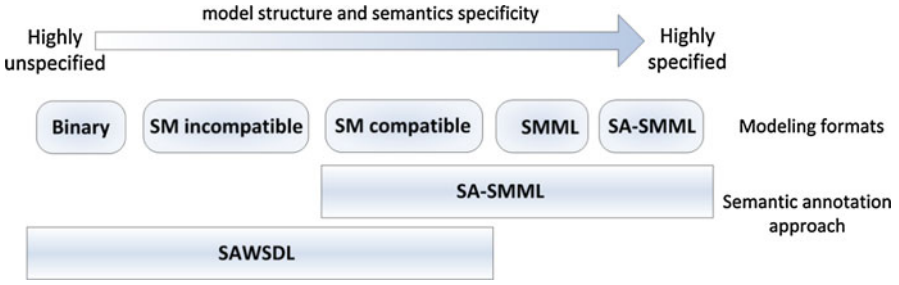

Fig. 5 Different types of models and their semantic annotation approaches

from within the model representation format itself through the SA-SMML approach. Alternatively, if SM compatible models are available in the form of schema wrapper web services, the SAWSDL approach can be used to directly annotate them. Schema wrapper services provide operations to access and set the model parameters, and complete model schemas, and essentially treat models as 'data' exposed in the form of a web service. SM compatible models may also be available in the form of proxy web services, and the SAWSDL approach can be used to annotate these models as well. The distinction between the schema wrapper services and the proxy services is that the proxy services have model execution operations that can access model solver services to solve the model in addition to parameter access, essentially considering models as providing a 'service'. For example, a GAMS optimization model, exposed as a proxy service can invoke a GAMS model solver service to solve the model, whereas a schema wrapper service would be restricted to providing access to the model parameters and the model schema. Models incompatible with the SM paradigm, but represented in a higher level model representation format, can also be encapsulated in the form of proxy web services or schema wrapper services, and the SAWSDL annotation approach can be used for their semantic annotation and model delivery.

At another of the spectrum are models appearing in executable or binary formats with little to no structural and semantic details available about them. They can be termed as "black box" models, in comparison with other kinds of "white box" models which have higher structural and semantic specificity. These models are generally based on legacy code, and documentation is often sparse and restricted to required input and output data formats. They are encapsulated using wrapper services and can be annotated using the SAWSDL approach.

In sum, various modeling language and delivery formats have been accounted for in considering how such models may be semantically annotated. The following section demonstrates the use of SA-SMML to facilitate model discovery and composition in a distributed setting.

5 Demonstration

In this section, we revisit the motivational scenarios presented in Sect. 2.1, and demonstrate the utility of semantic annotation of decision models.

5.1 Model discovery

Referring to the patient admission forecasting scenario described earlier, an analyst interested in searching for a model to address the forecasting problem at hand engages in the model discovery process. During model discovery, search queries are formulated using URI(s) denoting the interested domain semantic concept(s) (e.g., *patients*, *prior_admission*, *physician_availability*). Querying for models annotated using SA-SMML can be done at two levels of granularity. At the low granularity level, semantic references for any of the following can be queried for: (a) model, (b) module, (c) genus, (d) elemental detail, (e) field, and (f) field value. Each model

in the resultant set is categorized as a precise match, or a partial match. If there is a complete mismatch between the queried ontology concept and relevant semantic references in the model registry, no resultant model is obtained. A precise match is further categorized as 'equivalent' or 'generalized.' An equivalent match indicates that the requested domain concept is the same or equivalent to the semantic reference term in the retrieved model. A generalized match indicates that the queried ontology concept (e.g., *patients*) is subsumed by the resultant model's semantic reference (e.g., *healthcare_participants* genus), indicating the resultant model refers to a more general concept than desired. A partial match indicates that the resultant model's semantic reference (e.g., *patients* genus) is subsumed by the queried ontology concept (e.g., *healthcare_participants*), indicating the resultant model refers to a more specific concept than desired.

At the high granularity level, a query consists of a match pattern that represents the desired pattern of semantic concepts (Ludwig and Reyhani 2006; Gomadam et al. 2009). Such patterns may be created for model schemas as well as model instances. Figure 6 shows a skeleton of match patterns for model schemas and model instances indicating in the ovals the type of semantic reference that may be provided. Based on this skeletal structure, Fig. 7 shows (a) a sample match pattern for a model schema for finding a weighted average forecasting model for predicting patient admissions, and (b) a corresponding model retrieved from the model repository. The sample match pattern shown consists of a set of semantic concepts for the model as well as the contained modules and genera. Matching of the patterns to retrieve models can be done in a hierarchical manner (Gomadam et al. 2009; Ludwig and Reyhani 2006). In other words, for model schemas, the semantic references are searched in the following order: model, modules, genera. At each step, a match may be equivalent, generalized, or partial. An overall match score for each of the resultant models can be computed using a weighted scheme. The search queries are ultimately performed based on the inferences derived by a reasoning engine (e.g., Pellet, Racer, Jess) operating on the domain semantic models, i.e. ontologies.

5.2 Model composition

Now, consider the model composition scenario described in Sect. 2.1. In this case, an analyst is interested in dealing with a more complex decision problem having to do with using prior patient admission data to generate staffing schedule for nurses. As such, a model composition approach is warranted, given that model discovery will not result in a single model that addresses the decision problem. A number of model composition approaches exist in the literature, both in service composition as well as model management areas, which can be leveraged and built upon for this purpose. In absence on semantic information, however, these approaches inherently assume that the models in the model base use compatible domain terminology, and thus rely solely on syntactic matching of key terms in the model base to satisfy the constraints (in the form of inputs, outputs, etc.) presented in the model composition requirements. The semantics incorporated in the SA-SMML representation allows model parameter matching based on subsumption reasoning of ontological concepts

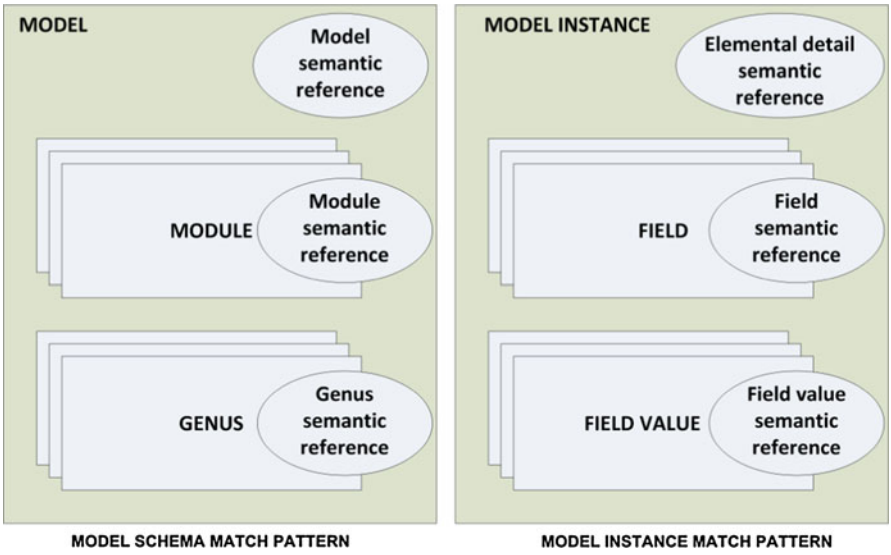

Fig. 6 Match pattern skeleton for querying SA-SMML models

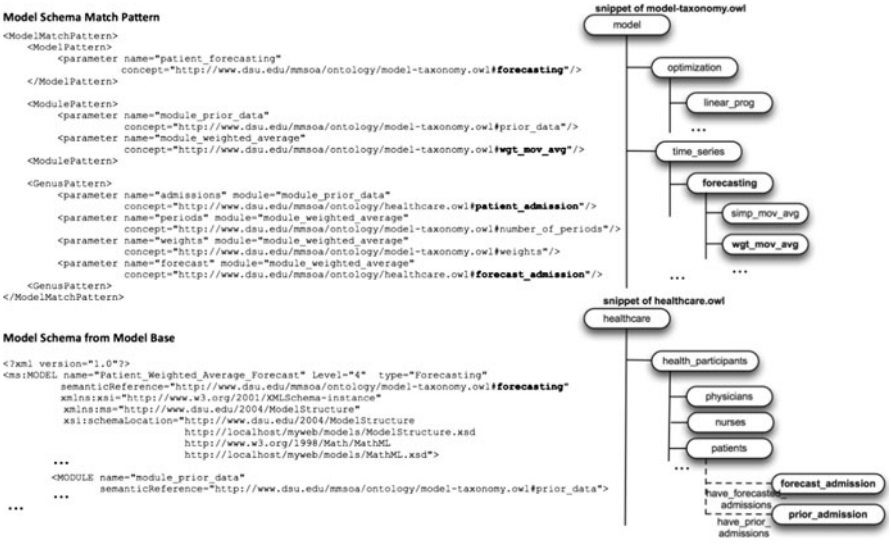

Fig. 7 A model schema match pattern and a matching model schema

and can thus provide composition results (equivalent, generalized, and partial matches), not possible through keyword matching. An example of this is shown in Fig. 8 where model genera *forecast_admission* and *desired_nurse_staff_level* are matched semantically. Different design choices exist in how semantic matching may be integrated with model composition. One possible choice entails performing semantic search on the results obtained through syntactically model composition.

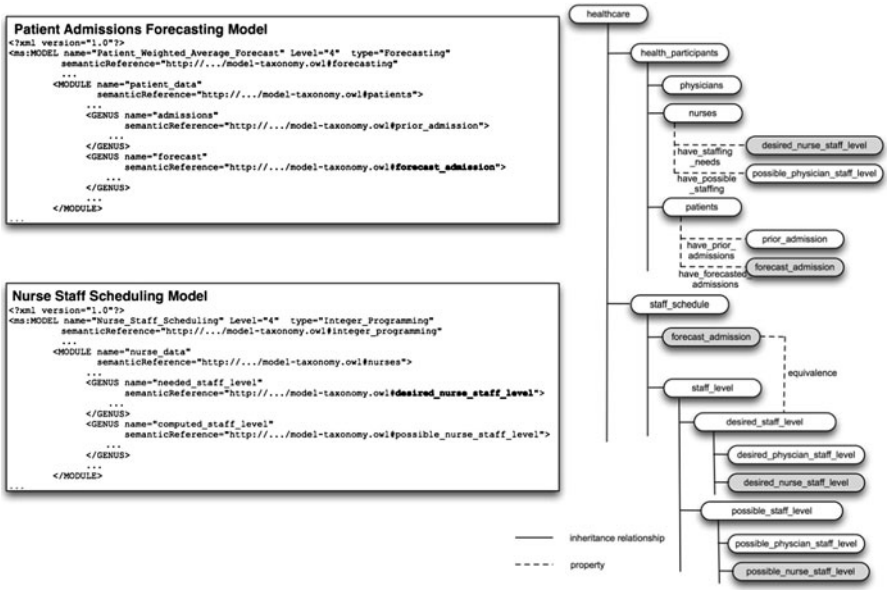

Fig. 8 Semantic matchmaking for model composition

This approach is useful in filtering out any model compositions that have been resulted due to syntactic similarities, but semantic dissimilarities. However, the disadvantage is that models that may be syntactically dissimilar, but semantically similar are not considered in the matching process. Another design choice is to perform semantic matching during each stage of the model composition-matching algorithm, thus closely coupling both matching techniques. This approach, while producing better results, can be computationally intensive depending the underlying model composition technique used. It is beyond the scope of this paper to discuss the integration of semantic matching with model composition techniques.

6 Evaluation and discussion

Following the demonstration of the utility of the SA-SMML model representation approach, we now evaluate the proposed approach in terms of how it has addressed each of the requirements discussed in Sect. 3. SA-SMML is an extension of SMML, which in turn is founded on the SM representation. The advantage of using SM as a conceptual foundation is that it has been shown to meet the basic model representation design requirements of model-paradigm independence, model-data independence, model-solver independence, and it captures meta-model information to allow computational reasoning. SMML additionally provides a standards-based (XML) common model representation format, while maintaining extensibility (such as the SA-SMML extension). The XML-based approach also has the advantages of

being vendor-neutral and interoperable with different development environments and information systems.

The proposed SA-SMML extension is focused on incorporating semantic information within decision models, which is accomplished by providing pointers to semantic models from within SMML. The design requirements for semantic annotations are met by SA-SMML as follows.

- *Machine-understandability*: The semantic information incorporated in SA-SMML is in the form of pointers to semantic models expressed as ontologies in knowledge representation formats like OWL. Computational reasoning capabilities available to operate on these semantic knowledge representation formats ensure machine-understandability of the semantics.
- *Semantic interoperability*: Semantic information encoded within the model schema and instances facilitate semantic interoperability by using same semantic references to point to common concepts. Also, reasoning engines can use the semantic models to infer whether the intended semantic concept is equivalent, generalized or partial match of what is desired.
- *Expressivity*: The semantic models in the form of ontologies used to annotate SA-SMML models reside outside of the models themselves and thus their expressivity in capturing the domain knowledge is governed by the underlying logical formalism adopted. Description logics-based OWL ontologies are more expressive than RDF and RDFS.
- *Inferencing support*: With regards to inferencing, the expressivity of the ontologies will dictate their inferencing capabilities. However, it is noted that for OWL-DL ontologies, reasoning engines such as Pellet, Racer, Fact++ can perform inferencing to draw requisite inferences. As such, SA-SMML satisfies the requirement of inferencing support to the extent that the referenced semantic models are expressive and support inferencing.
- *Leveraging existing efforts and standards*: The SA-SMML proposal leverages prior model representation efforts in developing SMML and also uses the SAWSDL standard to ensure interoperability with service enterprise systems.
- *Ease of application development*: The use of standards-based approach leads to ease of application development using SA-SMML, given that Application Programming Interfaces (APIs) for dealing with WSDL, XML schema conversions, and semantic models (e.g., OWL) are readily available through community open source efforts.

The use of SA-SMML model representation scheme has several managerial and technical implications. From a technical standpoint, it can be noted that while the proposed approach demonstrated the semantic annotation approach for decision models by extending SMML, arguably other kinds of XML-based languages can also be extended to incorporate such semantic information. The criteria for such extensibility would be access to XML schema for the concerned modeling language and development of tools to parse and interpret the semantic references encoded. On a related note, the use of SMML as an intermediary format is contingent upon development of model translator services to different formats. Future work will involve extensive development of such model translator services. Incorporating

semantics is dependent on the availability and accessibility of relevant ontologies. Potential challenges associated with the development of ontologies that are widely accepted by the user community can impede such effort. From a managerial standpoint, a scheme such as SA-SMML for annotation decision models provides an opportunity for managers to leverage the current investments in modeling resources such as modeling platforms, solvers, etc. while providing an ability to operate in a distributed environment, based on a SOA. In the same vein, organizations can benefit through sharing and reuse of modeling resources, and thus avoiding duplication of efforts. Facilitating model sharing, however, can have ownership, intellectual property, and security implications that will need to be addressed. Incentives, organizational policies, and contractual obligations may mitigate (as well as facilitate) model sharing.

7 Conclusion

In summary, this article proposes an XML-based model representation approach, called SA-SMML, extending the SMML by including links to semantic concepts at relevant places in the model schemas and instances. The article demonstrates the utility of the proposed SA-SMML approach in the context of model querying and composition. Compared to other modeling approaches, this approach has the distinct advantage that it allows inferences to be drawn upon rich problem domain semantics, along with the model structure, while performing model management functionalities like model search and discovery, model selection, and model composition. Further, the packaging of models as services afford them to be used seamlessly in a scalable manner as part of a SOA. Last, but not least, the approach is accommodative of multiple modeling languages and formats and the design decision of conceiving models as services is useful in wrapping other kinds of models as services as well and adding semantic information using a standards-based WSDL approach.

References

Basu A, Blanning RW (1994) Metagraphs: a tool for modeling decision support systems. Manage Sci 40(12):1579–1600

Bhargava HK, Kimbrough SO (1993) Embedded languages for model management. Decis Support Syst 10(3):277–299

Bhrammanee T, Wuwongse V (2008) ODDM: a framework for modelbases. Decis Support Syst 44(3):689–709

Blanning RW (1985) A relational framework for join implementation in model management. Decis Support Syst 1:69–85

Blanning RW (1993) Model management systems: an overview. Decis Support Syst 9(1):9–18

Bose R, Sugumaran V (2007) Semantic web technologies for enhancing intelligent DSS environments. In: Decision support for global enterprises. pp 221–238

Bradley GH (2003) Introduction to eXtensible markup language (XML) with operations research examples. INFORMS Comput Soc Newslett. http://www.nps.navy.mil/orfacpag/resumePages/papers/Bradley-INFORMSComputingNewletterMay2003.htm. Accessed 23 Nov 2011

Bradley GH (2005) Network and graph markup language (NaGML)—data file formats. In: Ninth INFORMS computing society conference, Maryland, January 5–7 2005. INFORMS
Brandes U, Eiglsperger M, Herman I, Himsolt M, Marshall M (2004) GraphML. http://graphml.graphdrawing.org/. Accessed December 6, 2006
Brooke A, Kendrick D, Meeraus E (1988) GAMS: a users guide. The Scientific Press, Redwood City
Bunke H (1982) Attributed programmed graph-grammars and their applications to schematic diagram interpretation. IEEE Trans Pattern Anal Mach Intell 4:574–582
Chang A-M, Holsapple CW, Whinston AB (1993) Model management issues and directions. Decis Support Syst 9(1):19–37
Chari K, Sen TK (1998) An implementation of a graph-based modeling system for sturctured modeling (GBMS/SM). Decision Supp Syst 22(2):103–120
Charnes A, Cooper W, Rhodes E (1978) Measuring the efficiency of decision making units. Eur J Oper Res 2(6):428–449
Chase RB, Apte UM (2007) A history of research in service operations: What's the big idea? J Oper Manag 25(2):375–386
Choobineh J (1991) SQLMP: a data sublanguage representation and formulation of linear mathematical models. ORSA J Comput 3:358–375
Consortium for Service Innovation (2007) http://www.serviceinnovation.org/
Data Mining Group (2010) Predictive model markup language (PMML). http://www.dmg.org/. Accessed September 15, 2010
Demirkan H, Kauffman R, Vayghan J, Fill H, Karagiannis D, Maglio P (2008) Service-oriented technology and management: perspectives on research and practice for the coming decade. Electron Commer Res Appl 7:356–376. doi:10.1016/j.elerap.2008.07.002
Dolk DR (1988) Model management and structured modeling: the role of an information resource dictionary system. Commun ACM 31(6):704–718
El-Gayar OF, Tandekar K (2007) An XML-based schema definition for model sharing and reuse in a distributed environment. Decis Support Syst 43:791–808
Erl T (2004) Service-oriented architecture: a field guide to integrating XML and web services. Prentice Hall PTR, Upper Saddle River
Ezechukwu OC, Maros I (2003) OOF: open optimization framework. Department of Computing, Imperial College London
Ferguson DF, Stockton ML (2005) Service-oriented architecture: programming model and product architecture. IBM Syst J 44(4):753–780
Fourer R (1983) Modeling languages versus matrix generators for linear programming. ACM Trans Math Softw 9(2):143–183
Fourer R, Gay DM, Kernighan BW (1993) AMPL: a modeling language for mathematical programming. The Scientific Press, Redwood City
Fourer R, Lopes L, Martin K (2005) LPFML: a W3C XML schema for linear and integer programming. INFORMS J Comput 17(2):139–158
Fourer R, Ma J, Martin K (2006) OSiL: an instance language for optimization. department of industrial engineering and management sciences. Northwestern University, Chicago
Gagliardi M, Spera C (1997) BLOOMS: a prototype modeling language with object oriented features. Decis Supp Syst 16:1–21
GAMS Development Corporation (2010) The GAMS model library index. http://www.gams.com/modlib/modlib.htm. Accessed September 15 2010
Geoffrion AM (1987) An introduction to structured modeling. Manage Sci 33(5):547–588
Geoffrion AM (1989a) The formal aspects of structured modeling. Oper Res 37(1):30–51
Geoffrion AM (1989b) Reusing structured models via model integration. Paper presented at the proceedings of the twenty-second hawaii international conference on system sciences (HICSS-22 '89), Kailua-Kona, HI
Geoffrion AM (1992a) The SML language for structured modeling: levels 1 and 2. Oper Res 40(1):38–57
Geoffrion AM (1992b) The SML language for structured modeling: levels 3 and 4. Oper Res 40(1):58–75
Glushko R, Tabas L (2009) Designing service systems by bridging the "front stage" and "back stage". Inform Syst e-Bus Manag 7(4):407–427
Gomadam K, Ranabahu A, Wu Z, Sheth AP, Miller J (2009) A declarative approach using SAWSDL and semantic templates towards process mediation. In: Petrie C, Margaria T, Lausen H, Zaremba M (eds) Semantic web services challenge: results from the first year, Springer-verlag, Berlin, pp 101–118

Heffner R (2011) SOA adoption 2010: still important, still strong. Forr Res. http://www.forrester.com/rb/Research/soa_adoption_2010_still_important%2C_still_strong/q/id/59058/t/2?src=RSS_CustomFeed&cm_mmc=Forrester-_-RSS-_-Document-_-9. Accessed 23 Nov 2011

Holocher M, Michalski R, Solte D, Vicuña F (1997) MIDA: an open systems architecture for model-oriented integration of data and algorithms. Decis Support Syst 20(2):135–147

Holt R, Schurr A, Elliott S, Winter A (2002) Graph eXchange Language. http://www.gupro.de/GXL/. Accessed December 6, 2006

Huh SY (1993) Model based construction with object oriented constructs. Decis Sci 24(2):409–434

Jones CV (1990) An introduction to graph based modeling system: part 1. Overview. ORSA J Comput 2(2):180–206

Jones CV (1991) An introduction to graph based modeling systems, part 2. Graph grammars and their implementation. ORSA J Comput 3(3):180–206

Katz S, Risman LJ, Rodeh M (1980) A system for constructing linear programming models. IBM Syst J 19(4):505–520

Kim H (2001) An XML-based modeling language for open interchange of decision models. Decis Support Syst 31:429–445

Kopecký J, Vitvar T, Bournez C, Farrell J (2007) SAWSDL: semantic annotations for WSDL and XML schema. IEEE Internet Comput 11(6):60–67

Krishnan R, Chari K (2000) Model management: survey, future research directions and a bibliography. Interact Trans OR/MS 3(1). http://www.informs.org/Pubs/ITORMS/Archive/Volume-3/Volume-3-No.-1-Krishnan-and-Chari. Accessed 23 Nov 2011

Kristjansson B (2002) Optimization modeling in distributed applications: how new technologies such as XML and SOAP allow OR to provide web services. http://www.maximal-usa.com/slides/Montrl02/index.htm. Accessed February 20, 2006

Lenard ML (1986) Representing models as data. J Manag Inform Syst 2(4):36–48

Lenard ML (1993) An object oriented approach to model management. Decis Support Syst 9:67–73

Liang T-P (1985a) Integrating model management with data management. Decis Support Syst 1(3):221–232

Liang TP (1985b) Integrating model management with data management in decision support systems. Decis Support Syst 1(3):221–232

Liang TP, Konsynski BR (1993) Modeling by analogy: use of analogical reasoning in model management systems. Decis Support Syst 9:113–125

Lopes L, Fourer R (2005) XML-based proposals for optimization. senna.iems.nwu.edu/xml/. Accessed June 17, 2005

Ludwig SA, Reyhani SMS (2006) Context-aware ontology selection framework. In: Ontologies: a handbook of principles, concepts and applications in information systems. Integrated series in information systems, vol 14. Springer, Berlin, pp 607–634

Machuca JAD, del Gonzalez-Zamora MM, Aguilar-Escobar VG (2007) Service operations management research. J Oper Manag 25(3):585–603

McLean CR, Lee TY, Riddick FH, Shao G (2005) Shop Data model and interface specification. National institute of standards and technology (NIST), NIST Interagency/Internal Report (NISTIR)—7198

Metters R, Marucheck A (2007) Service management—academic issues and scholarly reflections from operations management researchers. Decis Sci 38(2):195–214

Miller LW, Katz N (1986) A model management system to support policy analysis. Decis Support Syst 2(1):55–63

Muhanna WA (1993a) An object oriented framework for model management and DSS development. Decis Support Syst 9(1):217–229

Muhanna WA (1993b) An object-oriented framework for model management and DSS development. Decis Support Syst 9(2):217–229

Muhanna WA (1994) SYMMS: a model management system that supports reuse, sharing, and integration. Eur J Oper Res 72:1093–1123

Muhanna WA, Pick RA (1994) Meta-modeling concepts and tools for model management: a systems approach. Manage Sci 40(9):1093–1123

Mukhopadhyay S, Samaddar S (2007) Improving revenue management decision making for airlines by evaluating analyst-adjusted passenger demand forecasts. Decis Sci 38:309–327

Paschalidis IC, Tsitsiklis JN (2000) Congestion-dependent pricing of network services. IEEE ACM Trans Netw 8(2):171–184

Peffers K, Tuunanen T, Rothenberger MA, Chatterjee S (2007) A design science research methodology for information systems research. J Manag Inform Syst 24:45–77
Roth AV, Lenor LJ (2003) Insights into service operations management: a research agenda. Prod Oper Manag 12(2):145–164
Sage AP, Armstrong JE Jr (2000) An introduction to systems engineering. Wiley, New York
Sahai A, Graupner S (2005) Web services in the enterprise: concepts, standards, solutions, and management. Springer, New York. doi:10.1007/0-387-27597-5
Stohr EA, Tanniru M (1980) A database for operations research models. Int J Policy Anal Inform Syst 4(1):105–121
Tien JM (2003) Toward a decision informatics paradigm: a real-time, information-based approach to decision making. IEEE Trans Syst Man Cybernet Part C Appl Rev 33(1):102–113
Wang Y-H, Lu Y-C (2002) An XML-based DEVS modeling tool to enhance simulation interoperability. In: Verbraeck A, Krug W (eds) 14th European simulation symposium, 2002. SCS Europe BVBA
Wright PD, Bretthauer KM, Cote M (2006) Reexamining the nurse scheduling problem: staffing ratios and nursing shortages. Decis Sci 37(1):39–70
Zeigler BP (1990) Object-oriented simulation with hierarchical modular models: intelligent agents and endomorphic systems. Academic Press, Boston