2012

An ontology-based model management architecture for service innovation

Omar F. El-Gayar
Dakota State University

Amit Deokar
Dakota State University

Recommended Citation

El-Gayar O.F., Deokar A.V. (2012) An Ontology-Based Model Management Architecture for Service Innovation. In: Dolk D., Granat J. (eds) Modeling for Decision Support in Network-Based Services. Lecture Notes in Business Information Processing, vol 42. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-27612-5_7

An Ontology-Based Model Management Architecture for Service Innovation

Chapter *in* Lecture Notes in Business Information Processing · January 2012

DOI: 10.1007/978-3-642-27612-5_7

2 authors:

Omar F. El-Gayar
Dakota State University

Amit Deokar
University of Massachusetts Lowell

An Ontology-Based Model Management Architecture for Service Innovation

Omar F. El-Gayar and Amit V. Deokar

College of Business and Information Systems, Dakota State University
820 N. Washington Avenue, Madison, SD 57042 USA
{omar.el-gayar,amit.deokar}@dsu.edu

Abstract. Organizations have indicated renewed interest in service innovation, design and management, given the growth of service sector. Decision support systems (DSS) play an important role in supporting this endeavor, through management of organizational resources such as data and models. Given the global nature of service value chains, there have been ever increasing demands on managing, sharing, and reusing these heterogeneous and distributed resources, both within and across organizational boundaries, through DSS consisting of database management systems (DBMS) and model management systems (MMS). Analogous to DBMS, model management systems focus on the management of decision models, dealing with representation, storage, and retrieval of models as well as a variety of applications such as analysis, reuse, sharing, and composition of models. Recent developments in the areas of semantic web and ontologies have provided a rich tool set for computational reasoning about these resources in an intelligent manner. In this chapter, we leverage these advances and apply service-oriented design principles to propose an ontology-based model management architecture supporting service innovation. The architecture is illustrated with case study scenarios and current state of implementation. The role of potential information technologies in supporting the architecture is also discussed. We then provide a roadmap to make advancements in research in this direction.

Keywords: Services, Services Innovation, Model Management, Ontologies, Service-oriented Architecture.

1 Introduction

The past two decades have witnessed a significant shift in emphasis from traditional manufacturing to service sectors for major economies in the world. Service sector industry verticals such as healthcare, business consulting, and education employ more than 80% of the workforce in the United States [1]. Similar trends have been observed globally with over 20% growth in the service sectors in other countries like Japan, France, Italy, China, and India [2-3]. Traditional manufacturing-based organizations are expanding to become service giants, realizing that beyond reengineering and maintaining efficient processes, service-related capabilities can enhance their business

D. Dolk et al. (Eds.): Decision Support Modeling in Service Networks, LNBIP 42, pp. 143–168, 2012.

models by providing huge service revenues. For instance, IBM's targeted services have undergone expansion from 23% in 1992 to more than 52% in 2006 [4].

Given this pervasiveness of services, much attention is being devoted towards understanding and managing service systems. At a foundational level, service systems are characterized as value-creation networks situated in organizational contexts utilizing people, technology, and informational resources in different magnitudes [2, 5]. However, given that the capabilities in today's emerging services are significantly different from traditional production and service capabilities, the business transformations in developing and managing modern day service systems is a challenging endeavor [3, 6]. Recent industry efforts such as IBM's Service Science, Management, and Engineering (SSME) initiative [5], and initiatives led by consortiums such Service Research and Innovation Initiative (SRII) [7] and Consortium for Service Innovation (CSI) [8] are noteworthy here. Essentially, the focus is on creating service innovations with measurable outcomes.

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. Efforts from these disciplines are geared towards providing better decision models to enable effective and efficient decision making. 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.

Further, information technology (IT) has opened doors to many new opportunities, including providing new services electronically as well as innovating traditional services through use of IT, leading to increased collaborative efforts in distributed environments. However, in practice, 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. Thus, in the context of service enterprises, the need for distributed decision support in general, and model management in particular is more today than ever before [9]. Additionally, existing model representations are not directly amenable to architectures supporting distributed environments. Last, but not least, such representation schemes are often paradigm dependent. In effect, without a scalable architecture for managing models in distributed environments that captures the structure and semantics of models as well as preserves model-data, model-solver, and model-paradigm independence, efforts to support "modeling in the large" and to leverage existing investments in models through sharing and reuse are seriously curtailed. In this chapter, we propose an architecture for model management in a heterogeneous and distributed environment. The architecture leverages recent development in service oriented computing, web services, and the semantic web to facilitate model sharing and reuse in such environment and illustrate how such architecture can serve as an enabler for service innovation.

The remainder of the chapter is organized as follows. The following sections presents a brief review of services and service innovation followed by the role of decision models in services innovations and a number of motivating scenarios highlighting the need for a system facilitating the management of models on heterogeneous environments. Next, we provide a detailed description of the proposed model management architecture, and an illustration of the utility of such architecture using a number of representative scenarios for model management. The final section summarizes the main contributions of the proposed architecture and highlights directions for future research.

2 Services and Services Innovation

2.1 The Concept of Services

The services sector truly began to show its presence on the economic front beginning in the 1970s. Given that these initial developments preceded the rise of information technology advances, the services under consideration then were the brick-and-mortar service enterprises (such as restaurants, and traditional banking), what are now termed as "traditional" services, in which the physical environment of service interaction received primary focus.

Definitions of the term "service" abound. Lovelock [10] characterizes services along several dimensions: its nature, the relationship with the client, decisions, economics, and mode of delivery. Fitzsimmons and Fitzsimmons [11] defines service as "a time-perishable, intangible experience performed for a customer acting in a role of co-producer". Gronroos [12] defines services as "processes consisting of a series of activities where a number of different types of resources are used in direct interaction with a customer, so that a solution is found to a customer's problem." Lusch [13] defines service as "the application of specialized competences (knowledge and skills), … for the benefit of another entity or the entity itself". Implicit in these definitions is the recognition that decision models are key resources utilized throughout the service management lifecycle.

With the advances of information technology (IT), attention is shifting towards "network-based services" also referred to as "emerging services" [3]. Compared to traditional services, these services are differentiated by the fact that they are IT-driven, have low labor requirements, emphasize self-service and high transaction speed, leverage data from multiple heterogeneous sources, are computationally intensive, and focus on mass customization [3]. Online auctioning such as eBay, e-commerce and targeted marketing such as Amazon.com, self-service travel sites such as Travelocity and Expedia are just but few examples of network-based services. Evangelista [14] identify a similar class of services termed as “science and technology based services”, where the firms are engaged in innovation activities such as R&D, engineering and technical consultancy including computing and software services. Organizations involved with such services are originators of new technological know-how, which is then diffused to manufacturers and other services. The service activities identified here are located upstream in the innovation and knowledge generation chain, where the goal is to provide appropriate solutions to a variety of information and technical needs and requirements of clients, exploiting available technologies. Harnessing decision support technologies to support these novel breed of services is essential.

With the increasing advances in technology, it can be argued that IT has a critical role to play in supporting the service value chain [5]. Network-based services are essentially dependent on IT for co-creating the value, as well as providing the competitive edge for the provider organization [15] . The role of technology is seen to be twofold. On one hand, technology is dramatically changing the way services are created, designed, and delivered. Research is progressing towards infusing and integrating technology in service encounters such as e-commerce, and self-service systems [16]. On the other hand, technology has a role to play in providing decision support in managing and delivering services and service related artifacts. Artifacts such as decision models underpinning service encounters need to be effectively utilized within organizations and even with partnering organizations across the boundaries. Research in this area is sparse and worthy of more attention [17].

2.2 Service Innovation

Given the ubiquity of services, the area of innovation in services is drawing increasing attention among scholars. The term “innovation” in the context of services has primarily two connotations. Service innovation can imply introduction of a new service offering, analogous to “product innovation”, or the development of a new way of managing or delivering a service, analogous to “process innovation”. The former can be considered “demand-side” driven innovation with emphasis on growing revenue through market expansion and meeting customer requirements, while the later can be considered “supply-side” driven innovation with emphasis on reengineering and/or improving service management for higher productivity [18]. A strong connection between these two apparently disjoint (strategic vs. operational) views can be noted from an overall service management lifecycle perspective.

Two distinct aspects of service innovation are evident from the literature: organizational innovation and technological innovation [19]. Research along the lines of organizational innovation has focused on developing conceptual tools and models to depict the peculiarities of services such as intangibility, and the highly social nature of interactions [20-21]. On the other hand, research in the area of technological innovation focuses on technological advances and ways in which they can innovate services [22-23]. Thus, it is interesting to recognize multiple and complementary perspectives adopted in studying innovation in services. An example of such emerging perspectives is the notion of "open innovation" or "distributed innovation", which recognizes the interactive nature of modern innovation processes, within and across the service systems [24-25]. It underscores the need to support innovators forming coalitions by sharing their knowledge [26].

Regardless of the definition of services, decision models are an important resource and knowledge objects that underpin any successful service venture and are utilized throughout the service management lifecycle. Recent developments in IT and associated IT-enabled service innovation further emphasizes the increasing role of decision models in providing decision support in managing and delivering services and service related artifacts and in enabling service innovations. The following section elaborates the role of decision models in supporting services and service innovations and highlights the significance of managing these models as an organizational resource.

3 Role of Decision Models in Service Innovation

Krishnan and Chari [27] depict a model (or a model schema) as a formal abstract representation of a decision problem. In other words, models can be conceived as specific formulations of decision situations amenable to certain problem solving techniques, such as simple linear regression, or an LP product mix formulation [28]. 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 generate model solutions.

Decision models have played a major role in the evolution of manufacturing process and systems towards just-in-time manufacturing and mass customization. Examples of such models were employed in production planning and distribution, facility design, inventory control, and total quality management. With much of the research in operation management (OM) traditionally dominated by manufacturing issues [29], initial efforts focused on adapting and utilizing such models in service design and management. As OM research on services evolved, increased recognition of the distinct characteristics of services and 'breaking free' from the goods-based manufacturing perspective became apparent [29-30]. Along these lines, decision models continued to evolve into models specifically catering to service needs such as yield management and customer selection. Other work emphasized the distinction and similarities between manufacturing and service systems to identify problems (and associated solutions) such as portfolio optimization, workforce optimization, and resource allocation that pertain to both systems [31].

Recent developments in information technology (IT) have been key enablers of service innovations referred to as "emerging services". Examples of such service innovations that are highly dependent on IT include sectors such as financial services and banking, retail, and tourism. Technologies such as Radio Frequency Identification (RFID) and Universal Product Code (UPC) have revolutionized services such as retail and transportation of goods. However, such emerging service innovations enabled by IT have also created challenges with respect to the handling and processing of large amounts of data for decision making (often in real-time), creating what Tien [32] 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 DRIP problem noted above [32]. 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. The research described in this paper builds upon the notion of decision informatics, particularly from a model management standpoint, in supporting the service innovation process.

To better understand the role of decision models in service innovation, we adopt a systems engineering perspective on services [3]. In this perspective, a service system life cycle is composed of the following phases adapted from [33]:

- *Need assessment/Requirements and specification*: The objective is the identification of user requirements and the translation of these requirements into specification for service and supporting processes.
- *Design/Development*: This involves the design of various aspects of the service, e.g., number of servers, delivery and communication mechanisms, etc.
- *Service production and delivery*: In contrast to goods, service production and delivery is co-located in time and space.
- *Service evaluation and optimization*: Following production and delivery, the service is evaluated based on performance measures.
- *Phaseout/Disposal*: A service system may be phased out or replaced by another service based on the results.

Decision models can be used in various phases along the service system life cycle. For example, demand forecasting models can be used in need assessment, while workforce and service portfolio optimization can be used in the design of service innovations. Real-time yield management models may be used in services such as hotels and airlines. Data Envelopment Analysis (DEA) may be used for evaluating service productivity and provide the basis for further innovations as depicted in Table 1.

Table 1. Examples of decision models/application by service system life cycle phase

Service system life cycle phase	*Examples of decision models/applications*
Need analysis	Portfolio optimization (determining the scope, scale and composition of services) and demand forecasting, e.g., [34]
Service design	Queuing models for determining server configurations, e.g., [35]; Workforce scheduling (may also be at the time of delivery, i.e., in real time)
Service production and delivery	Yield (revenue) management, e.g., [36], workforce optimization (workforce level, composition, and assignment), e.g., [37]
Service evaluation and optimization	Data envelop analysis (DEA) [38]

Alternatively, we can view the role of models along Schmenner's Service Process Matrix [39] which distinguishes among various service industries by the degree of labor intensity and the level of interaction and customization involved. Regardless of service type, decision models have been productively employed to address distinct service issues as shown in Table 2.

Models can also be viewed as knowledge objects encapsulating an organization's knowledge about a decision problem in a particular domain. The CSI [8] 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 [27] comprised of problem identification, model creation, model implementation, model validation, model solution, model interpretation and model maintenance represent 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, recent emphasis on agile business processes and workflows, particularly in the context of service innovation is a major driver for service-oriented computing and architecture. By viewing models as services (as will be described later) within an enterprise service oriented architecture, models can provide the necessary analytics and decision support in real-time to the flexible configuration and re-configuration of business processes and workflows further enabling service innovation.

Table 2. Examples decision models by industry type adapted from Schmenner [39] and Rust [40]

		Interaction and customization	
		Low	**High**
Labor intensity	**Low**	***Service factory*** Example of industries – Airline – Hotels – Trucking Example of models – Yield management	***Service shop*** Example of industries – Hospitals – Repair services Example of models – Layout and queuing analysis
	High	***Mass Service*** Example of industries – Retailing – Retail banking Example of models – Data envelop analysis (DEA)	***Professional services*** Example of industries – Doctors – Lawyers – Stockbrokers Example of models – Customer selection models

In summary, the following observations are made about the role of models in service innovation:

– Decision models have been developed and used for the design, delivery and management of services. Nevertheless, the ubiquity of IT and the resulting abundance of data further underscore the significance of the development and application of such models in supporting various service life-cycle phases.
– The increased use of software agents as enablers for service innovation [41] coupled with the availability of decision models in machine-readable form (as web services) further expands the support for service innovation.
– Models as knowledge objects encapsulate knowledge involved in the design, delivery, and management of services. The management of these models as a part of an organizational management strategy further supports service innovation as advocated by the CSI [8]
– Decision models as web services support agile business process and service innovation.

In essence, the aforementioned discussion underscores the need for managing models as an organizational resource and a key enabler for service innovation. The following section presents representative scenarios further motivating the need for model management in the context of service innovations followed by a brief review of model management literature highlighting major contributions as well as limitations as enablers of service innovation.

4 Motivating Scenarios

4.1 Scenario 1: Intra-organizational Model Sharing

Consider an organization offering its services in diverse markets, both regionally and internationally. The headquarters uses a marketing simulation model for developing the best marketing mix including the advertising expenditure, product quality index, and product distribution. This model is used in analyzing different what-if scenarios regarding marketing strategies and their effect on the organization's market share and volume. Another branch operating in markets characterized by high variability in demand, different competition conditions, and distribution channels would need to reuse the former model. Reusing such a model will involve customizing the model to take into consideration new parameters and competitive dynamics for this branch's market. Alternatively, various branches may have developed their own models or have adapted existing models to meet their specific requirements. It would be advantageous if each such branch is able to share its models with other branches (including the headquarters) in a seamless manner.

Unfortunately, in practice, such goal is often hampered by lack of awareness of the existence of such models in the first place, heterogeneity of modeling environments resulting in accessibility and compatibility issues, and inadequate (or lack of) documentation that often employs inconsistent semantics complicating the problem of assessing the applicability of a particular model as well as the possibility of customizing such models to the situation at hand. The aforementioned issues are even more prevalent in an inter-organizational setting.

4.2 Scenario 2: Models as Knowledge Objects

Knowledge intensive business services (KIBS) are private organizations that rely heavily on professional knowledge for supplying intermediate products or services that are knowledge based [42]. Examples of KBIS include IT support services, management consultancy, and engineering consultancy. According to Hertog [43], KIBS capture scientific and technological information that is often dispersed across the economy, and tailor such information to meet the needs of its client. KIBS can be considered as catalysts and co-producers of innovation [43]. Interaction between KIBS and their clients involve extensive knowledge flows which take a variety of forms. While tacit knowledge is a significant component of knowledge flow, explicit forms of knowledge such as written reports, project plans, software, and decision models are also prevalent.

In this scenario, consider a management consultancy firm specializing in helping client firms answer questions pertaining to their projected energy demand, cost, and optimal mix of their energy portfolio. The firm relies on a number of decision models for forecasting energy demand and supply, prices for various forms of energy, transportation and distribution costs, etc. A client wishes to use the services of the consultancy firm, specifically the client is interested in integrating the energy price forecasting models developed by the consultancy firm with its own production and distribution models. Given the likelihood that the client may be using different modeling environments and assumptions, such integration may be severely hampered. The situation is further complicated if the client wishes to select and test a variety of such models for their suitability to their particular needs. Such a situation may be encountered with other clients.

4.3 Scenario 3: Model Management Supporting Service Design and Agile Business Processes

The design of a new service frequently involves the use of decision models such as queuing models, resource planning and allocation models, and service portfolio optimization. While such models are often developed in the context of a new service, over time, such models may be applied to other service innovations. For example, consider a parcel delivery company that is experimenting with innovative ways for routing and delivery of packages. In this case, existing routing models may be adapted to reflect various routing configurations. While such a scenario is plausible and feasible using current technology, a number of situations may arise that would constrain such possibilities, namely,

- Decision makers or analysts may not be aware of the existence of such models in the organization in the first place,
- Models may have been developed and implemented using obsolete technologies and no longer accessible,
- Models may lack the documentation necessary for proper utilization.

A model repository in a broader context of a model management system in a heterogeneous environment will help alleviate such situations. The following section highlights contributions and limitations of related work on model management.

5 Related Work on Model Management

Model management (MM) encompasses a variety of functionality including model description, model manipulation, scheduling, execution, and information display. Research in the area of model management emerged since the 1980s in the context of managing models in decision support systems (DSS). Management science and operations research applications provided a fertile environment for this interest. While a comprehensive review of the model management literature can be found elsewhere [28, 44], it is worth noting that much of the motivation behind model management focused around finding ways for developing, storing, manipulating, controlling, and effectively utilizing models in an organization [45]., Some of the important developments are noted below, along with highlighting the need for distributed model management.

In general, models can be seen to conform to a modeling lifecycle, consisting of a complex, iterative process during which several modeling tasks are accomplished.

Some of the modeling tasks are computationally intensive, while others are more subjective and need human judgment and domain expertise. Supporting the modeling life-cycle entails providing a number of functionalities. For example, model creation may involve description, formulation, selection, integration, composition, and reuse of models. The need for providing more expressive power in describing models has driven the research on explicit model representations using meta-modeling techniques such as Structured Modeling (SM) [46]. While model formulation focuses on the knowledge elicitation involved in the development of new models, the remaining steps in model creation aim at leveraging repositories of existing models. 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 is an important component of model management in the decision support context, where decision models are desired to be composed from individual model units. It is often used interchangeably with the term model integration in the literature. However, we try to distinguish between the two terms based on the approach taken for synthesizing models. Model integration focuses on synthesizing models at the structural or definitional level [47-49]. At this level, different model schemas are integrated in a cohesive manner. Model composition, on the other hand, focuses on assembling models at a functional level [50-52]. Model implementation is concerned with issues related to creating model representations amenable to execution by solvers, with focus on model-data, model-solver, and model-paradigm independence. Post-solution model interpretation deals with issues facilitating the interpretation of results by modelers and decision makers, such as the analysis of the sensitivity of model results to parameter variations, the analysis of the sensitivity of the model to structural changes in the model, and the inspection of model structure.

Most of the MM research since the early 1980s up to the mid-1990s focused on addressing these functionalities and requirements of MM systems. However, over the past decade and a half, additional requirements concerning portability, vendor independence, and compatibility have become critical due to the feasibility of sharing models within and across organizations driven by advances in supporting communication infrastructure. With the exception of Muhanna [9] and few others, very little attention was paid to managing large shared model bases. Accordingly, a major limitation of the aforementioned approaches is their limited support to the requirements for model sharing in a distributed environment. It has thus become critical to meet the increased globalization demands in today's service economy.

Over the past decade, Distributed Model Management Systems (DMMS) have emerged as a new breed of information systems engaged in distributed model management activities such as model creation and delivery, model composition, model execution and model maintenance to fulfill dynamic decision-support and problem solving requests. Bhargava, Krishnan, and Muller [53] propose a web-based architecture for sharing decision models, illustrated using the DecisionNet prototype. The purpose of DecisionNet is to provide decision support technologies accessible electronically to consumers as a service over the World Wide Web instead of being purchased as stand-alone products. In this sense, DecisionNet performs the role of an "agent," mediating transactions between consumers and providers, in essence a "yellow pages" of services. Dolk [54] proposes an integrated modeling environment that utilizes structured modeling for representing models, data warehouses for storing models, and a component-based architecture for plugging in software components based on user needs. Huh, Chung and Kim [55] propose a framework for collaborative model management systems in a distributed environment. The emphasis is on coordinating the changes made to a collection of shared models and propagating the effect of these changes throughout the organization. In the context of optimization models, Ezechukwu and Maros [56] propose an architecture supporting distributed optimization over the Internet.

To facilitate the distribution process of model management, Web services pose as a viable technology to accomplish the mediation task. Web services are based on service-oriented computing principles and provide a standardized way of integrating several application modules using open standards such as XML (Extensible Markup Language), SOAP (Simple Object Access Protocol), WSDL (Web Services Description Language) and UDDI (Universal Description, Discovery, and Integration) over an Internet protocol backbone. In that regard, Iyer, Shankaranarayanan, and Lenard [57] propose a model management architecture emphasizing the use of structured modeling to represent spreadsheet models and an architecture supporting the sharing and reuse of models and data. Also, Madhusudan [58] proposes an architecture in which a "service platform" acts as a mediator by accepting service requests (e.g., from decision support clients), composing applicable models, and orchestrating the execution of the models. Supporting model representation in a web services environment, Kim [59] and El-Gayar and Tandekar [60] propose XML-based representations for analytical models. Both languages are based on the structured modeling paradigm [46] for conceiving, manipulating, and representing models at a higher level of abstraction to facilitate drawing inferences about models.

According to Goul and Corral [61], enterprise modeling refers to the activities, process representations and conceptualizations of an enterprise. The objective is to improve enterprise integration and support analysis of an enterprise. Such models are more likely to exist as a collection of models rather that one monolithic model [62]. As envisioned by Ba et al. [62], a critical element of an enterprise modeling framework is the ability to automate building and executing task-specific models (from existing model fragments) as needed in response to user generated requests. Recent work by Sen, Demirkan, and Goul [63], further extends this notion by proposing an architecture for dynamic and inter-organizational decision support.

In this chapter, we propose an architecture for the management of decision models in a heterogeneous and distributed environment, by building upon previous work on DMMS noted above as well as leveraging recent advances in service-oriented and semantic web technologies. The proposed architecture complements Sen et al.'s [63] architecture at layer 2 "Unified Enterprise Modeling Language (UEML) representation of models" and layer 3 "Decision Support Environment (DSE) middleware" with a focus on models supporting decision making processes in a service enterprise.

6 An Architecture for Distributed Model Management

6.1 Design Considerations

A number of design issues guided the formulation of the proposed architecture. These issues pertain to general DSS requirements for distributed support and to specific requirements for distributed model management. In that regard, we identify the following desirable features and design characteristics of a modeling system [9, 46, 56]:

- a conceptual framework for modeling based on a single model representation format,
- representational independence of model structure and the detailed data,
- representational independence of model structure and the model solution,
- meta-modeling capability to support reasoning about models,
- extensibility for different modeling paradigms, and
- accessibility of decision support resources.

6.2 Models as Services

Conceptually, a model as a loosely coupled component delivering a specific functionality can be conceived as a (computational or web) service. Likewise, a service as an entity abstracting underlying logic can be considered as a model. In reference to the aforementioned principles underlying service orientation [64], and in the context of model management, the following is noted:

- *Reuse*: Much of the work underlying model selection, composition, and integration focuses on finding ways to leverage existing models through reuse.
- *Abstraction*: A model is an abstraction of reality. To facilitate model selection and composition, models commonly expose only the models' description and interface. Note that model integration with its underlying 'white box' assumption is inconsistent with service-oriented principles.
- *Autonomy*: Similar to services, within its boundary (execution environment), models have complete autonomy independent of other models.
- *Loose coupling*: Related to abstraction and autonomy, and in the context of model selection and composition, models are loosely coupled with other models.
- *Statelessness*: Models exhibit statelessness, thereby supporting loose coupling and autonomy characteristics.
- *Composability*: Supporting reusability, models may be composed from other models, and may also participate in the composition of other models.
- *Discoverability*: Models should facilitate their description and discovery for consumption by other models.

In effect, with the exception of model integration and model interpretation, a significant synergy exists between model management, and service-oriented technologies and management. Together, models and model management functionalities wrapped as computational services form the components of the architecture. The proposed architecture highlights opportunities for synergy between these two arenas.

6.3 Distributed Model Management Architecture

The proposed architecture for distributed model management systems builds on earlier work on distributed decision systems, with a particular emphasis on model management, and is illustrated in Figure 1. At the core of the architecture is a service bus, which provides the underlying communication infrastructure for various model management services. The bus supports intra and inter-organization communication among services by implementing web services standards such as SOAP over HTTP. Connected to the bus is a collection of decision support services such as infrastructure management services, user interface services, and model management services. These services provide access to a variety of decision support resources such as models, modeling environments, and solvers. A decision support component acting as a client, can access any of the services connected to the bus irrespective of the physical location of the service. To facilitate intra and inter-organizational communication among services, the architecture adopts XML web services in which all services communicate via Internet protocols (mostly HTTP) and all data messages are sent and received as XML documents.

In this architecture, infrastructure management services may include discovery services for registering (publishing services), and services for configuring, monitoring and operating services. Account management services provide software licenses and access to fee-based services. Adopting XML web services, discovery services are implemented as Universal Description, Discovery, and Integration (UDDI) server managing information about all registered services, i.e., serving as a registry. UDDI uses XML to represent its contents, and contain enough information to direct clients to resources outside it such as web services description language (WSDL) files, which in turn provide information about the functionality of a service and the details necessary to communicate with the service.

Data needed for decision support is available in various formats and is often distributed. Data may reside in containers such as database management systems or data warehouses, or may reside as stand-alone files. Most contemporary database management systems provide support to XML. With the data provider and wrapper services registered with the discovery services, a client component which may be a decision support application or another decision support service would be able to locate and access the desired data. These data management services lie outside the scope of the current architecture; however the architecture is extensible to utilize such data management services.

User interface services are a collection of reusable and sharable components providing functionality for visualizing data and model results, customizing display and access for decision makers, and capturing user input for data and model management services. The presentation layer of a decision support system may use user-interface services as building blocks for developing the interface.

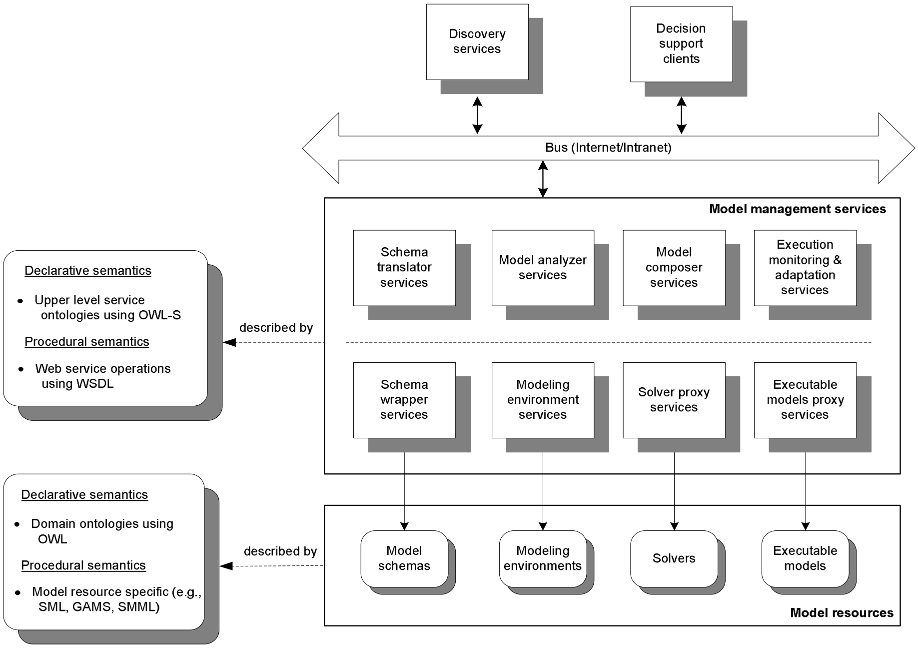

Fig. 1. A semantic web services-based architecture for model management systems

Model management services provide access and management to a variety of modeling resources. These resources include specialized solvers, modeling platforms and languages such as GAMS and AMPL, model schemas represented as text files, e.g., GAMS, and MATLAB models, and models as executable components. Different types of services are needed to utilize the various resources. For executable models and solvers, wrapper services are used to encapsulate the functionality of existing modules as web services. For development environments and platforms such as AMPL or MATLAB, proxy services are used to expose the functionality and to manage the interface with these environments. For model schemas represented as stand-alone non-executable files such as GAMS, AMPL, and XML representations of models, schema wrapper services encapsulate the functionality and purpose of the underlying models and coordinate the interface with other services to successfully execute these models.

Other model management services include services for model translation, model composition, and model analysis. Model translation services represent a repository of services for translating model schemas to/from a variety of popular formats and languages such as GAMS and LINGO to/from SMML. Model composition services in association with the execution monitoring and adaptation services allows for leveraging a collection of models for a specific decision situation by coordinating the execution of such models in a workflow-like manner. Model analyzer services provide functionality for analyzing model results and conducting what-if analysis.

The architecture facilitates service innovation by supporting different phases of the service system life cycle (see Section 3). The needs assessment and specification phase includes knowledge encoding leading to the formulation of specific requests for model retrieval from an existing modeling repository or new model formulations applicable to a specific service context. The service design and development phase includes harnessing discovery services in tandem with model management services depicted in the architecture. The service designer can also utilize other related services such as user interface services, and decision support clients, for facilitating new service design. Model composer services can computationally generate compatible sequences of existing models that can meet the user requirements. These models may be simulated or used in real time, depending upon the life cycle stage. The execution monitoring and adaptation services can guide the service production and delivery phase, while the model analyzer services can help in the service evaluation and optimization phase. Other associated model management services such as schema translator and wrapper services, modeling environment services, solver proxy services, and executable model proxy services can be invoked to conduct the low level model management tasks. Finally, the phase out stage is guided by the results of the model analyzer services to determine the relevance and currency of the model for a particular service application context.

The architecture is distributed in the true sense, in that even the model management functionalities are exposed as web services. This is contrary to many distributed model management approaches discussed in the previous section, where although model resources are distributed, the model management functionalities reside in a centralized manner (e.g., see [58]). This approach has a major advantage in that a decision maker can query, compose, or deploy models using only a thin client, without bearing the burden of model management computations.

Further, two main aspects of the architecture are evident for model sharing and reuse. First, semantic description of models using OWL, provides a mechanism to reason about their properties. Second, the semantics associated with OWL-based ontologies of models is extensible for describing corresponding web services using OWL-S. These semantic web services can then be used either as atomic model units or composed together into composite model units to derive a solution for a particular decision making problem. Since both atomic as well as composite services are described using OWL-S, they can be discovered more effectively through logic-based search techniques, rather than just keyword based search. These two novel aspects of model management are discussed below.

6.4 Semantic Descriptions of Models

Ontologies are explicit conceptualizations (i.e., meta-information) that describe the semantics of information resources. Significant advances have recently been made along the lines of using ontologies for reasoning about resources available on the Web [65] . Model resources such as models, solvers, or executable models that are distributed over the Web can lend themselves to these advances to provide better model management capabilities, particularly in distributed settings. In this section, we discuss some of the relevant advances and standards that are instrumental in realizing the semantic web infrastructure, and their particular application in the context of the proposed model management architecture.

The Resource Description Framework (RDF) is a W3C standard that builds on top of XML to provide a data model for describing resources on the Web in terms of named properties and values, and encoded in a formal, machine-processable format [66]. An RDF description of a resource consists of a set of RDF statements (or triples). Each RDF triple consists of three parts: an object (a resource), an attribute (a property), and a value (another resource or plain literal). RDF Schema (RDF-S) extends RDF by providing a type system for RDF or an ontological vocabulary for describing properties and classes of RDF resources [66]. RDF-S thus provides a way to build an object model with a semantics for generalization-hierarchies of such properties and classes. The Web Ontology Language (OWL) goes a step further by adding more vocabulary for describing properties and classes [66]. Some examples include property type restrictions, equality, property characteristics, class intersection, and restricted cardinality.

Web services are a class of resources that are distributed, similar to models. In the earlier discussion, we built an analogy between models and services. Essentially, web services are self-describing, self-contained software applications that are accessible over the Internet [64]. Web services form a cornerstone of our proposed architecture for model management systems and its relation to semantic web technologies is discussed next.

Currently, models as web services are described procedurally using the Web Services Description Language (WSDL), which lack semantic descriptions of web services [67]. Several research approaches have been proposed to add semantics to web service descriptions [68]. Four submissions 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) [69]. Recently, W3C put forth a modified version of WSDL-S, Semantically Annotated WSDL (SAWSDL) as their recommendation [70]. Analysis of these standards revealed OWL-S to be more amenable to model management, given that it defines its meta-model in the same language that it uses for concrete service descriptions as well as allows for more expressive languages to be incorporated with OWL. OWL-S is hence selected for providing semantics to models, encapsulated as web services in our architecture, and is briefly discussed here. It can also be noted that significant synergy exists between OWL-S and SWASDL, and that research is being conducted on grounding OWL-S in SAWSDL, i.e., relating elements of an OWL-S service description with elements of a WSDL service description [71].

OWL-S is an OWL-based Web Service Ontology language, whose objective is to provide a vocabulary for encoding rich semantic web service descriptions, in a way that builds upon OWL [69]. Service descriptions may be provided using OWL-S that mainly consists of three interrelated sub-ontologies for the top-level concept Service, namely service profile, service model, and service grounding. The service profile is used to express ‘what a service does’, which may be used for service advertising, constructing service requests, and matchmaking. The service model provides essentially a process model to describe 'how the service works', in the form of inputs, outputs, preconditions, and effects (typically called IOPE), which may be used for service seeking, composing service descriptions, coordinating and monitoring of service executions. However, it can be noted that OWL-S takes the view that a process is not necessary a program to be executed, but a specification of the ways in which a client may interact with the service. There can be three types of processes: atomic, composite, and simple. Atomic processes correspond to the actions a service can perform by engaging it in a single interaction; composite processes correspond to actions that require multi-step protocols and/or multiple server actions; and simple processes provide an abstraction mechanism to provide multiple views of the same process. Finally, the service grounding provides information on 'how the service can be accessed' by mapping the constructs of the process model onto detailed specifications of message formats, protocols, and so forth (typically expressed in WSDL).

Associating semantic metadata to models and other model resources is essential in order to reason about their capabilities. Above mentioned advances in ontologies and semantic web standards facilitate the provision of semantics to model resources through descriptions encoded in the form of their respective *domain ontologies*. Similarly, their corresponding web services as well as other supporting web services, such as model schema translator services, are described in the form of *higher level ontologies*, particularly designed for web services.

Semantic descriptions of models can facilitate multiple uses. They can provide metadata for intelligent searching, browsing, and composing of models by decision makers. Moreover, implicit assumptions, model uses, constraints, and such can be explicitly captured through creation of domain ontologies of models. In fact, certain models may be elaborated in detail to provide, what can be termed as a 'white box' representation of models. Models (or model schemas) described using paradigms such as structured modeling, e.g. using SMML [60], may be semantically expressed to the finest level of detail with domain ontologies. Certain additional elements such as the application domain, model purpose, and so forth, are standardized across every model described with an OWL-based domain ontology. Last, but not the least, these OWL-based domain ontologies can be extended to create higher level ontologies with OWL-S, in order to describe models wrapped as web services. The composer and execution monitoring services can make use of these higher level service ontologies to facilitate model composition functionalities.

Shown below is a snippet of domain ontology for a revenue computation model, described with OWL.

```
...
<owl:Class rdf:ID="FinancialModel">
   <rdfs:comment>Used to compute revenues and
                 Income
   </rdfs:comment>
   <rdfs:subClassOf rdf:resource="#Model"/>
</owl:Class>
```

```
<owl:DatatypeProperty rdf:ID="hasOutput">
   <rdfs:domain rdf:resource="#Model">
   <rdfs:range
            rdf:resource="&xsd;positiveInteger">
</owl:DatatypeProperty>
...
```

Similarly, shown below is a snippet of the OWL-S service ontology for the revenue computation model.

```
...
<Description rdf:about="#FinancialModel">
    <hasPreCondition>
  <expr:HTN-expression>
       <expr: expressionBody>
     ((computed-price ?product_price)
      (forecasted-demand
                                 ?product_demand)
      (production-cost ?pcost)
      (distribution-cost ?dcost))
       </expr: expressionBody>
  </expr:HTN-expression>
    </hasPreCondition>
</Description rdf:about="#FinancialModel">
```

In the following sections, we discuss the implementation of the proposed architecture, followed by illustrative case studies indicating the utility of this architecture in facilitating service innovation.

7 Implementation and Current Status

A research prototype of the proposed architecture has been developed using the J2EE platform. Key model management functionalities include model discovery services, model wrapper and translation services, modeling environment services, solver proxy services, and executable model proxy services. These have been extended based on our prior research effort. The current emphasis is on developing model composer and execution monitoring services. The goal is to utilize the application context encoded in the OWL and OWL-S domain ontologies to semantically extract candidate models that may satisfy the model composition request. The semantic descriptions of models, provided in the form of higher level service ontologies using OWL-S, serve as a building block in how the model composer service may function. The model library is populated with associated ontologies in OWL and OWL-S using the Protégé editor. One of the important developments underway is the ability to provide computational translation of models described using different representation techniques into OWL and OWL-S model ontologies, similar in notion to the work on dynamic decision support by Sen, et al. [63].

8 Case Studies

To demonstrate interaction of the various services comprising the proposed architecture, we have developed a series of Unified Modeling Language (UML) sequence diagrams. Two representative case studies are discussed below.

8.1 Case study 1: Model Sharing in an Inter- or Intra organizational Setting

This case study emphasizes the efficacy of the proposed architecture in addressing some of the issues most frequently encountered in sharing models in an inter- or intra-organizational setting, such as model awareness and the heterogeneity of modeling environments.

In this case study (Figure 2), a decision support client (DSC) wishes to identify a decision model for the problem at hand, e.g., a contract portfolio optimization model or a scheduling model (as described in the motivating scenarios). In the architecture, the DSC may use a discovery service to locate an appropriate model (thereby addressing the awareness issue) and data (if necessary). In this case study, the model happened to be represented in the Structured Modeling Markup Language (SMML), a XML-based representation for mathematical models [60]. This is an abstract representation of model structure that is not suitable for direct execution. Accordingly, when the DSC requests the execution of the model, the SMML portfolio optimization model proxy service uses a translator service to translate the SMML model into a representation such as the General Algebraic Modeling Language (GAMS). To be able to solve the GAMS model, the model proxy server uses the discovery service to locate a proxy service of the respective environment to process the model. The environment proxy service may then use the discovery service to locate an appropriate solver and execute

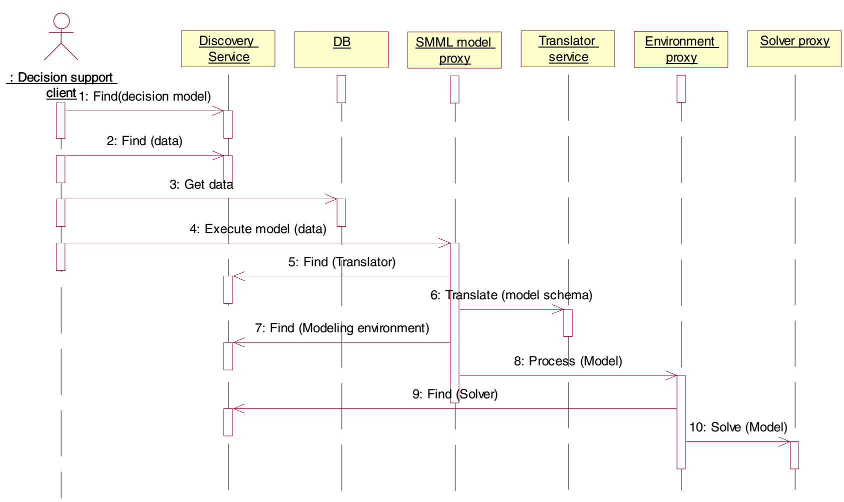

Fig. 2. Case study 1: Model sharing in an inter- or intra-organizational setting

the solver to solve the model. In this case study, a decision maker/analyst would be able to meet his/her decision needs regardless of whether the decision maker (or decision support tool) is familiar to the particular model representation or has access to the modeling environment and solvers necessary to solve the model.

8.2 Case study 2: Model Composition in an Inter-organizational Setting

This case study (Figure 3) demonstrates a typical interaction among model management services for composing a model from existing models and executing them in the appropriate order. An example of such scenario may occur in the interaction between a KIBS firm and its clients as described in the motivating scenarios where a client would wish to integrate its problem specific models with models provided by the KIBS.

In this case study, the decision support client (of the KIBS client) uses the discovery service to first try and locate the desired model. Due to unavailability of such a model, the decision support client then invokes a model composer service with the model request. The model composer service, in turn retrieves the semantic descriptions (OWL-S ontologies) of model resources (bundled as web services). Based on the service profile, and service model descriptions in these ontologies, the composer service extracts the form of inputs, outputs, preconditions, and effects (typically called IOPE, and may be used for service seeking, composing service descriptions, coordinating and monitoring of service executions) for each model resource. In the prototype implementation, Hierarchical Task Network (HTN) planning, which is a class of AI planning algorithms is used to search the state space for potential composition of available model resources to respond favorably to the model composition request. For a detailed example, the readers are referred to [72]. Then, the client uses the execution monitoring services to monitor the execution of the composite service generated. Semantic descriptions of selected model resources are retrieved by the execution monitoring service, since it uses the service model and service grounding descriptions in OWL-S ontologies to coordinate and monitor the ordered deployment of each model resource. In addition to addressing model awareness and heterogeneity issues, this case study demonstrates the use of model semantics to assess the applicability of models to a particular decision situation as well as to compose models (in heterogeneous environments) into new models that address specific needs of the decision maker.

As the ability to support service innovation becomes a major driver of success in service enterprises, facilitating model management for dynamic decision making throughout the service system life cycle becomes imperative. In this chapter, we have discussed the use of decision model management in facilitating service innovation in distributed service environments. The proposed distributed model management architecture is based on the confluence of service-oriented principles and semantic web technology. Design principles supported by SOA emphasize reuse, statelessness, autonomy, abstraction, discoverability, loose coupling, and composability. The proposed architecture for model management systems is novel, primarily in the

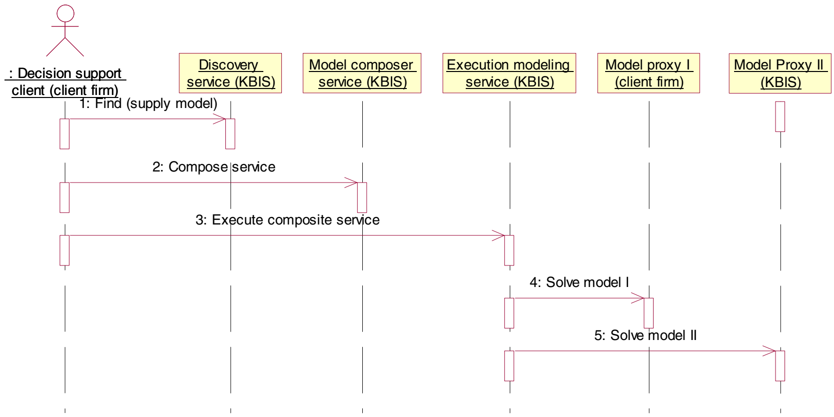

Fig. 3. Case study 2: Model composition in an inter-organizational setting

following two aspects. First, it is completely distributed through the provision for not only distributed model resources, but also distributed model management services. Second, it proposes a semantic layer for model representation that can facilitate automation and reasoning mechanisms, such as model composition. The utility of the architecture in supporting the different service management life cycle phases is highlighted. The motivating scenarios and case studies demonstrate several application areas where such an architecture may be best suited.

Further research is needed in how to reason about models across different service application contexts and industries. Enterprise ontologies have the potential of bringing together the needed semantics for adapting disparate models from different application contexts [61]. Also, model integration is noted to be important component along with model composition, focusing on synthesizing models at the structural (or definitional) level. This capability needs to be further studied in the light of semantic web technologies mentioned in this chapter. The application of model management in supporting agile business processes in service organizations has been noted earlier. However, this perspective needs further attention in terms of integrating workflow technology with model management techniques. Use of process ontologies in this regards seems to be a plausible avenue for further exploration [73]. Such related synergies are likely to create significantly new opportunities for global service organizations competing in an increasingly complex environment and striving for service innovation.

References

1. Hilsenrath, J.E.: The economy: Service sector enjoys growth, but pace slows. Wall Street Journal, New York, NY A.2 (2003)
2. Maglio, P.P., Srinivasan, S., Kreulen, J.T., Spohrer, J.: Service systems, service scientists, SSME, and innovation. Communications of the ACM (Special Issue: Services Science) 49, 81–85 (2006)

3. Tien, J.M., Berg, D.: A case for service systems engineering. International Journal of Systems Engineering 12, 13–39 (2003)
4. IBM: Annual Report (2006)
5. Spohrer, J., Maglio, P.P., Bailey, J., Gruhl, D.: Steps toward a science of service systems. Computer 40, 71–77 (2007)
6. Nambisan, S.: Why service businesses are not product businesses. MIT Sloan Management Review 42, 72–80 (2001)
7. Service Research and Innovation Initiative (2007), `http://forums.thesrii.org/srii`
8. Consortium for Service Innovation (2007), `http://www.serviceinnovation.org/`
9. Muhanna, W.A., Pick, R.A.: Meta-modeling concepts and tools for model management: A systems approach. Manage. Sci. 40, 1093–1123 (1994)
10. Lovelock, C.: Think before your leap in services marketing. In: Berry, L.L., Shostack, G.L., Upah, G. (eds.) Emerging Perspectives in Services Marketing, pp. 115–119. American Marketing Association, Chicago (1983)
11. Fitzsimmons, J.A., Fitzsimmons, M.J.: Services Management: Operations, Strategy, and Information Technology. McGraw-Hill, New York (2004)
12. Gronroos, C.: Service Management and Marketing. Wiley, Chichester (2000)
13. Lusch, R.F., Vargo, S.L. (eds.): The Service-dominant Logic of Marketing: Dialog, Debate, And Directions. M. E. Sharpe (2006)
14. Casati, F., Discenza, A.: Supporting workflow cooperation within and across organizations. In: Proceedings of the 2000 ACM Symposium on Applied Computing, pp. 196–202. ACM Press (2000)
15. Stauss, B., Engelmann, K., Kremer, A., Luhn, A. (eds.): Services Science: Fundamentals, Challenges and Future Developments. Springer, Berlin (2007)
16. Meuter, M.L., Bitner, M.J., Ostrom, A.L., Brown, S.W.: Choosing among alternative service delivery modes: An investigation of customer trial of self-service technologies. Journal of Marketing 69, 61–83 (2005)
17. Rust, R.T.: A call for a wider range of service research. J. Serv. Res. 6, 211 (2004)
18. Mansharamani, V.: Toward a Theory of Service Innovation: An Inductive Case Study Approach to Evaluating Uniqueness of Services. MIT Sloan School of Management, Cambridge, MA (2005)
19. Salter, A., Tether, B.S.: Innovation in Services: Through the Looking Glass of Innovation Studies. In: Research, A.I.o.M. (ed.) Background paper prepared for the Workshop Grand Challenge in Services, Oxford University, UK (2006), `http://www.sbs.ox.ac.uk/faculty/Sako+Mari/GCS.htm`
20. Sundbo, J., Gallouj, F.: Innovation as a loosely coupled system in services. In: Metcalfe, J.S., Miles, I.D. (eds.) Innovation Systems in the Service Economy: Measurement and Case Study Analysis. Kluwer Academic Publishers, Boston (2001)
21. Barras, R.: Towards a theory of innovation in services. Research Policy 15, 161–173 (1986)
22. Quinn, J.B.: Technology in services: Past myths and future challenges. In: Guile, B.R., Quinn, J.B. (eds.) Technology in Services: Policies for Growth, Trade and Employment. National Academy Press, Washington, D. C (1988)
23. Miles, I.: Services in the new industrial economy. Futures 25, 653–672 (1993)
24. Chesbrough, H.: Open Innovation: The New Imperative for Creating and Profiting from Technology. Harvard Business School Press (2003)

25. Coombs, R., Harvey, M., Tether, B.S.: Analysing distributed processes of provision and innovation. Industrial and Corporate Change 12, 1125–1155 (2004)
26. Brown, J.S., Duguid, P.: The Social Life of Information. Harvard Business School Press, Boston (2000)
27. Krishnan, R., Chari, K.: Model management: Survey, future research directions and a bibliography. Interactive Transactions of OR/MS 3 (2000)
28. Chang, A.-M., Holsapple, C.W., Whinston, A.B.: Model management issues and directions. Decision Support Systems 9, 19–37 (1993)
29. Metters, R., Marucheck, A.: Service management - Academic issues and scholarly reflections from operations management researchers. Decis. Sci. 38, 195–214 (2007)
30. Johnson, R.: Service operations management: Return to roots. International Journal of Operations & Production Management 19, 104–124 (1999)
31. Krishnamurthy, A.: From Just in Time Manufacturing to On-Demand Services. In: Hsu, C. (ed.) Service Enterprise Integration: An Enterprise Engineering Perspective, pp. 1–37 (2007)
32. Tien, J.M.: Toward a decision informatics paradigm: a real-time, information-based approach to decision making. IEEE Transactions on Systems, Man And Cybernetics: Part C. Applications And Reviews 33, 102–113 (2003)
33. Sage, A.P., Armstrong Jr., J.E.: An Introduction to Systems Engineering. John Wiley & Sons, New York (2000)
34. Iyer, A.V., Deshpande, V., Wu, Z.: A postponement model for demand management. Management Science 49, 983 (2003)
35. Borst, S., Mandelbaum, A., Reiman, M.I.: Dimensioning Large Call Centers. Operations Research 52, 17 (2004)
36. Paschalidis, I.C., Tsitsiklis, J.N.: Congestion-dependent pricing of network services. IEEE-ACM Transactions on Networking 8, 171–184 (2000)
37. Wright, P.D., Bretthauer, K.M., Cote, M.: Reexamining the nurse scheduling problem: Staffing ratios and nursing shortages. Decision Sciences 37, 39–70 (2006)
38. Charnes, A., Cooper, W., Rhodes, E.: Measuring the efficiency of decision making units. European Journal of Operational Research 2, 428–449 (1978)
39. Schmenner, R.: How can service businesses survive and prosper? Sloan Management Review Spring, 21–32 (1986)
40. Rust, R.T., Metters, R.: Mathematical models of service. European Journal of Operational Research 91, 427–439 (1996)
41. Tien, J.M.: Services Innovation: Decision Attributes, Innovation Enablers, and Innovation Drivers. In: Hsu, C. (ed.) Service Enterprise Integration: An Enterprise Engineering Perspective, pp. 39–76 (2007)
42. Miles, I., Kastrinos, N., Flanagan, K., Bilderbeek, R., den Hertog, P., Huitink, W., Bouman, M.: Knowledge intensive business services: Their role as users, carriers, and sources of innovation. EIMS Publication No. 15, Innovation Programme, DGXIII, Luxembourg (1995)
43. Hertog, P.D.: Co-producers of innovation: On the role of knowledge-intensive business services in innovation. In: Gadrey, J., Gallouj, F. (eds.) Productivity, Innovation and Knowledge in Services: New Economic and Socio-economic Approaches, pp. 223–253. Edward Elgar, Cheltenham (2002)
44. Blanning, R.W.: Model management systems: An overview. Decision Support Systems 9, 9–18 (1993)
45. Muhanna, W.A.: An object-oriented framework for model management and DSS development. Decision Support Systems 9, 217–229 (1993)

46. Geoffrion, A.M.: An introduction to structural modeling. Manage. Sci. 33, 547–588 (1987)
47. Basu, A., Blanning, R.W.: Model integration using metagraphs. Inf. Syst. Res. 5, 195–218 (1994)
48. Dolk, D.R., Kottemann, J.E.: Model integration and a theory of models. Decision Support Systems 9, 51–63 (1993)
49. Liang, T.-P., Konsynski, B.R.: Modeling by analogy: Use of analogical reasoning in model management systems. Decision Support Systems 9, 113–125 (1993)
50. Chari, K.: Model composition using filter spaces. Inf. Syst. Res. 13, 15–35 (2002)
51. Dhar, V., Jarke, M.: On modeling processes. Decision Support Systems 9, 39–49 (1993)
52. Kottemann, J.E., Dolk, D.R.: Model integration and modeling languages: A process perspective. Inf. Syst. Res. 3, 1–16 (1992)
53. Bhargava, H.K., Krishnan, R., Muller, R.: Decision support on demand: Emerging electronic markets for decision technologies. Decision Support Systems 19, 193–214 (1997)
54. Dolk, D.R.: Integrated model management in the data warehouse era. European Journal of Operational Research 122, 199–218 (2000)
55. Huh, S.Y., Chung, Q.B., Kim, H.M.: Collaborative model management in departmental computing. INFOR 38, 373–389 (2000)
56. Ezechukwu, O.C., Maros, I.: OOF: Open Optimization Framework. Department of Computing, Imperial College London (2003)
57. Iyer, B., Shankaranarayanan, G., Lenard, M.L.: Model management decision environment: a Web service prototype for spreadsheet models. Decision Support Systems 40, 283–304 (2005)
58. Madhusudan, T.: A web services framework for distributed model management. Inf. Syst. Front. 9, 9–27 (2007)
59. Kim, H.: An XML-based modeling language for the open interchange of decision models. Decision Support Systems 31, 429–441 (2001)
60. El-Gayar, O.F., Tandekar, K.: An XML-based schema definition for model sharing and reuse in a distributed environment. Decision Support Systems 43, 791–808 (2007)
61. Goul, M., Corral, K.: Enterprise model management and next-generation decision support. Decision Support Systems 43, 915–932 (2007)
62. Ba, S., Lang, K.R., Whinston, A.B.: Enterprise decision support using Intranet technology. Decision Support Systems 20, 99–134 (1997)
63. Sen, S., Demirkan, H., Goul, M.: Dynamic decision support through instantiation of UEML representations. Communications of the ACM 50, 87–93 (2007)
64. Papazoglou, M.P.: Web Services: Principles and Technology. Pearson-Prentice Hall, London (2008)
65. Fensel, D., Hendler, J., Lieberman, H., Wahlster, W. (eds.): Spinning the semantic web: Bringing the world wide web to its full potential. The MIT Press, Cambridge (2003)
66. 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 (2004), `http://www.w3.org/2001/sw/`
67. W3C: Web Services Description Language (WSDL) 1.1 (W3C Recommendation) (2001), `http://www.w3.org/TR/wsdl`
68. Sheth, A., Verma, K., Gomadam, K.: Semantics to energize the full services spectrum. Communications of the ACM (Special Issue: Services Science) 49, 55–61 (2006)

69. 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 (2005), http://www.w3.org/2002/ws/swsig/
70. Kopecký, J., Vitvar, T., Bournez, C., Farrell, J.: SAWSDL: Semantic Annotations for WSDL and XML Schema. IEEE Internet Comput. 11, 60–67
71. Paolucci, M., Wagner, M., Martin, D.: Grounding OWL-S in SAWSDL. In: Krämer, B.J., Lin, K.-J., Narasimhan, P. (eds.) ICSOC 2007. LNCS, vol. 4749, pp. 416–421. Springer, Heidelberg (2007)
72. Deokar, A.V., El-Gayar, O.F.: A semantic web services-based architecture for model management. In: Proceedings of the 41st Annual Hawaii International Conference on System Sciences (HICSS-41 2008). IEEE Computer Society, Waikoloa (2008)
73. Therani, M.: Ontology development for designing and managing dynamic business process networks. IEEE Transactions on Industrial Informatics 3, 173–185 (2007)