Decision-enabled dynamic process management for networked enterprises

Amit V. Deokar · Omar F. El-Gayar

Abstract Enterprises in today's networked economy face numerous information management challenges, both from a process management perspective as well as a decision support perspective. While there have been significant relevant advances in the areas of business process management as well as decision sciences, several open research issues exist. In this paper, we highlight the following key challenges. First, current process modeling and management techniques lack in providing a seamless integration of decision models and tools in existing business processes, which is critical to achieve organizational objectives. Second, given the dynamic nature of business processes in networked enterprises, process management approaches that enable organizations to react to business process changes in an agile manner are required. Third, current state-of-the-art decision model management techniques are not particularly amenable to distributed settings in networked enterprises, which limits the sharing and reuse of models in different contexts, including their utility within managing business processes. In this paper, we present a framework for decision-enabled dynamic process management that addresses these challenges. The framework builds on computational formalisms, including the structured modeling paradigm for representing decision models, and hierarchical task networks from the artificial intelligence (AI) planning area for process modeling. Within the framework, interleaved process planning (modeling), execution and monitoring for dynamic process management throughout the process lifecycle is proposed. A service-oriented architecture combined with advances from the semantic Web field for model management support within business processes is proposed.

Keywords Dynamic process management · Model management · Knowledge work

1 Introduction

Globalization and the emergence of a networked economy have resulted in numerous information management challenges for enterprises. Information technology (IT) has a vital role to play in addressing these challenges in the resultant complex and dynamic environments. From a process perspective, managing seamless flow of information within various activities in workflow processes is critical. Many of these processes cut across the boundaries of functional departments within organizations as well as organizational boundaries. Process-aware Information Systems (PAIS) such as Workflow Management Systems (WFMSs) and Enterprise Resource Planning (ERP) systems are aimed at supporting operational workflow processes, building on advances in information systems and technologies as well as management science (Dumas et al. 2005). Further, from a decision support perspective, many of the workflow activities in enterprises can benefit from analytical models of reality and vast amounts of data, which are used to realize key organizational objectives. Decision support systems (DSS) are focused on facilitating decision making through management of data and models. The basic thrust of such applications is to enable decision-makers to

A. V. Deokar (✉) · O. F. El-Gayar
Dakota State University,
Madison, SD, USA
e-mail: amit.deokar@dsu.edu

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

focus on making decisions rather than being heavily involved in gathering data, and conceiving and selecting analytical decision models. Efforts from management and decision sciences are thus geared toward coordinating knowledge workers' processes as well as providing relevant decision models needed to perform various activities, ultimately aiming at effective and efficient operations of the enterprise. While there have been significant advances in the areas of process management as well as decision sciences, there are significant challenges and research issues yet to be addressed.

One of key challenges facing organizations is that of synthesizing the decision and process perspectives in a cohesive manner, both from the viewpoints of conceptual modeling as well as technical feasibility. Second, although PAIS systems such as WFMS have indicated increases in business productivity and efficiency, their design has been limited to static and repetitive business processes, which are uncharacteristic of today's complex and dynamic environments. A more dynamic approach to providing process management support is thus needed. Third, the current state-of-the-art decision model management techniques are not particularly amenable to the distributed settings in modern networked enterprises. This limits the sharing and reuse of models in different contexts, including their utility within managing business processes.

In this paper, a framework that addresses the aforementioned issues is presented. The framework builds on computational formalisms, including the structured modeling paradigm for representing decision models, and hierarchical task networks from artificial intelligence (AI) planning area for process modeling. Within the framework, interleaved process planning (modeling), execution and monitoring for dynamic process management throughout the process lifecycle is proposed. Also, a service-oriented architecture combined with advances from the semantic Web field for model management support within business processes is proposed.

The paper is organized as follows. Section 2 discusses the process and decision modeling viewpoints and discusses key issues related to seamless integration of decision models in business processes. Section 3 presents a motivational scenario to provide a context and rationale for the paper. Next, Section 4 presents the requirements for the proposed framework as they relate to both dynamic process management as well as distributed model management. Following this, Section 5 presents the proposed framework in detail along with implementation notes. Section 6 illustrates the framework in the context of the motivational scenario identified earlier. Section 7 discusses the requirements in the context of the features provided in the framework. Finally, Section 8 concludes the paper.

2 Process and decision modeling perspectives

Traditionally, process modeling is concerned with representing, designing and optimizing business processes. According to Nilsson et al. (1999), business modeling refers to "the use of models and methods to understand and change business operations together with information systems in organizations". Examples of process modeling approaches include event-process chain (EPC) and extended event process chain (e-EPC) (Scheer 1999, 2000). Regardless of the modeling approach, it is paramount not only to capture process objectives, but also to couple these objectives with underlying organizational objectives (Neiger and Churilov 2006; Rolland and Prakash 2000). In that regard, goal-oriented business process modeling attempts to explicitly recognize business goals as a basis for identifying supporting activities and processes (Bider and Johannesson 2005). Alternatively, Neiger and Churilov (2006) propose a value-focused process engineering (VFPE) as a methodology to reconcile process-based objectives and organizational objectives. VFPE essentially integrates Keeney's (1994, 1996) value-focused thinking (VFT) decision modeling methodology with the extended event-driven process chain (e-EPC) process modeling methodology proposed by Scheer (2000). The resultant unified process modeling representation is termed as "decision-enabled EPC" (de-EPC) and provides a mapping of workflow patterns to objectives patterns that explicitly recognizes business objectives in business process design. Nevertheless, these approaches fall short of explicitly articulating how decision models (and tools) can support process objectives (which are ideally derived from business objectives).

On the other hand, decision modeling is concerned with the realization of organizational objectives through the development of descriptive and prescriptive models of a decision problem. While descriptive models, such as simulation models (Seila et al. 2003) allow for evaluating the consequences of various alternative courses of action under different contexts (inputs and processes), prescriptive models such as mathematical programming aim at providing optimal and feasible solution to a decision problem. Prescriptive models tend to be specific to highly structured decision problems and as a by-product, assist the decision maker in understanding the context and structure of the decision in terms of decision objectives, decision variables, and decision constraints. Despite the prevalence of these models in addressing a wide variety of decision problems, it is recognized (Neiger and Churilov 2002) that such models do not necessarily integrate seamlessly with existing business process.

In that regard, we advocate two issues that contribute to the lack of seamless integration of decision models in

existing business processes. First, from a modeling perspective, there is the likely potential for incompatibility of process and decision objectives. While, 'ideally' process models and decision models work towards the realization of common organizational goals, not all process models are driven by business objectives. On the other hand, decision models are not necessarily developed with a process in mind that would provide a holistic context for the application of the decision model. Second, from a technical perspective, decision models are developed using specialized tools and are represented in various forms complicating the potential for sharing and reusing such models in a workflow system. In essence, there is a need to explicitly capture the process and decision views of an activity as shown in Fig. 1, and to develop a supporting infrastructure that enable the seamless sharing and reuse of decision models in the context of supporting and executing business process.

In the following section, we present a motivating scenario highlighting the role of decision models in supporting business process, and the need for integration of the process and decision views. The example will also provide an appreciation of the potential technical issues confronted in designing a supporting workflow and model management system and a basis for outlining the requirements for such a system.

3 Motivational scenario—supply chain management

Supply chain management has received increased attention over the past decade and half and considerable efforts have been devoted to developing decision models for a variety of supply chain functions or activities (Narasimhan and Mahapatra 2004; Shapiro 2007; Swaminathan and Tayur 2003). This includes problem areas occurring at different levels of decision making. At the strategic level, examples include capacity planning (Paraskevopoulos et al. 1991), facility location (Revelle and Laporte 1996). At the tactical level, examples include bid selection (Talluri 2002), supplier selection (Narasimhan et al. 2003), and supplier evaluation (Talluri and Narasimhan 2003). Examples from the operational level include integrated operations (Cohen and Lee 1998) and procurement (Clark and Scarf 1960). It is imperative that the true value of these rich decision

models may be derived through their integration with business processes that underpin enterprise operations (Geoffrion and Krishnan 2003a, b). Recently, Bae and Seo (2007) illustrated the use of business process management systems for supply chain modeling, execution, and monitoring, with such an integrated viewpoint.

Figure 2 depicts a real-time operational planning example of order processing (Shapiro 2007). Delivery vehicles are to be scheduled and drivers' assignments to be made on a per-order basis, given short lead times. Linear programming (LP) and mixed integer programming (MIP) models can be employed to generate a routing solution and then re-optimize it with each new order. Driver assignment is a natural extension to such models. It may also be possible to generate a solution using heuristic algorithms with few initial orders, and then use the LP and MIP models to re-optimize the solution as orders increase. Thus, a variety of modeling alternatives are available to the knowledge worker completing the task. An operations research analyst would have likely analyzed different scenarios and developed a set of models to derive the solution and perform subsequent analysis. The model inputs and parameters will also likely vary from order to order and thus require a tailored solution. For example, the ordinary delivery hours might be 8 h, but this may need to be changed to 9 h if the order volume for a certain day is high or weather problems may be resulting in slower delivery times. Also, the model algorithm parameters such as the depth and breadth of the branch-and-bound search in solving an MIP problem may need to be tweaked for each optimization run. It can also be noted that Fig. 2 shows the process with high-level tasks that need to be further decomposed in smaller manageable tasks. For example, the optimal order routing task may be decomposed into granular tasks. One such task may involve the system recommending a set of models best suited for the decision problem based on knowledge of tasks completed thus far (such as figuring out whether a routing solution exists and needs re-optimization or not). The nature of the recommendation task will thus vary based on prior knowledge gathered. Other granular tasks may involve selecting the model schema from the recommended set of models, providing model parameters and inputs, executing the model, and fetching the results. This requires managing the process in a flexible and adaptive manner as well as

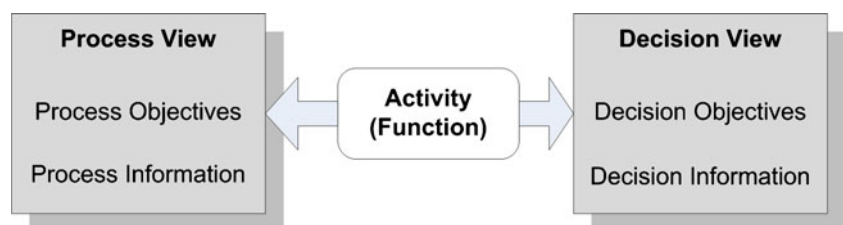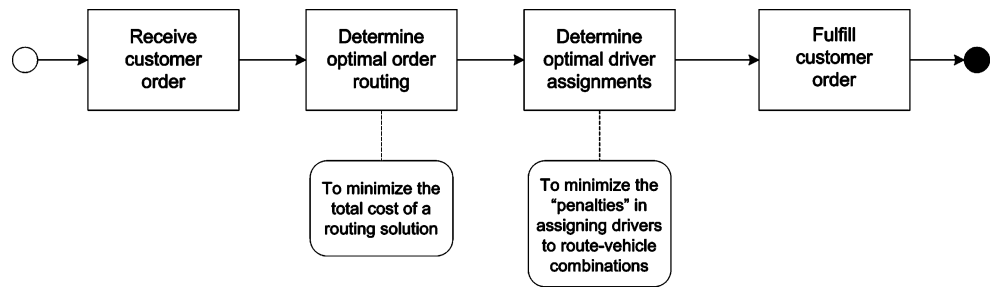Fig. 1 Process and decision views for modeling activities

Fig. 2 Order processing
example (Shapiro 2007)

supporting the decision-making tasks with relevant model resources.

4 Requirements for decision-enabled dynamic process support

Following the motivational scenario and the recognition of co-existence of decision and process perspectives, this section reviews relevant research in these areas and highlights key functional requirements for an integrated framework.

4.1 Dynamic process management

From an architectural viewpoint, PAISs play the role of abstracting the coordinative process logic from application logic. The process logic is derived based on core concepts in a process formalism or meta-model, e.g., process constraint language (Kumar and Zhao 1999), metagraphs (Basu and Blanning 1999), workflow nets (van der Aalst and van Hee 2002) at build time, while the process is orchestrated based on this process logic at runtime. While PAISs have shown significant promise in resultant productivity gains from efficient coordination of business processes, their applicabil-

ity in complex and dynamic organizational environments has been fairly limited. One of the prime reasons for this adoption issue is that the focus of most commercial PAIS systems such as workflow and business process management systems (BPMS) has been on coordinating and managing repetitive and structured business processes. Consequently, they provide minimal support for dynamic changes in business processes occurring in networked enterprises. Weber et al. (2009) recently surveyed issues that pertain to dynamic process lifecycle support, which are particularly noteworthy in this context and help guide the requirements for the proposed process management framework.

The process management lifecycle can be viewed as the one consisting of process (re)design, modeling, execution, and monitoring (refer Fig. 3). The technology infrastructure relates primarily to the later three phases, while the process (re)design is typically considered a pre-requisite managerial step involving translating business strategic goals into processes with specific process objectives. In turbulent environments, e.g., supply chain management (Trkman and McCormack 2009), process (re)design is highly knowledge-intensive and emergent. Complete articulation of every process choice and exception in process models is likely to be difficult in such cases. Weber et al. (2009) characterize

Fig. 3 Characteristics
of dynamic processes impacting
process lifecycle—adopted from
Weber et al. (2009)
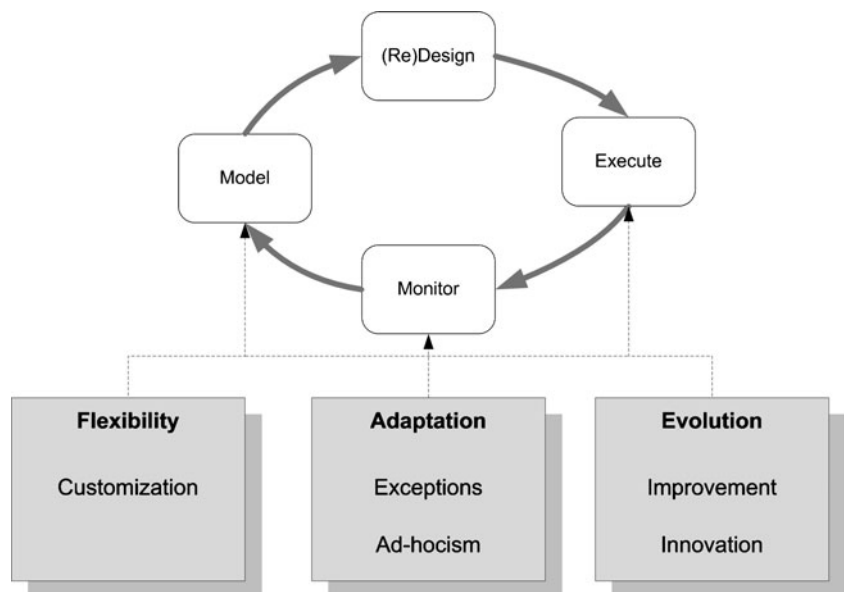
dynamic processes with respect to three main requirements, namely flexibility, adaptation, and evolution (refer Fig. 3). Flexibility refers to the ability to customize process instances based on their unique requirements at runtime by initially loosely or partially specifying the model at buildtime. Adaptation refers to the ability to handle anticipated or unanticipated exceptions. Evolution refers to ability of the process in place to change based on incremental or radical changes in the business process itself.

Soffer (2005) alternatively distinguishes between short term flexibility and long term flexibility depending on the extent of change in the process. Regardless, the impact of dynamic process on the process lifecycle is significant and has been a driver for several research initiatives in recent years. The reader is referred to Weber et al. (2009) for a comprehensive review of the same.

Based on the approaches identified in the literature for dealing with dynamic processes in agile environments as well as the main motivation of providing explicit support for decision-oriented tasks in a process, following design requirements are noted: (1) provide flexibility by allowing process instances to be customized, (2) support process evolution by allowing incremental or radical changes to processes, (3) adapt process instances by handling exceptions, (4) allow monitoring and traceability of process states, and (5) effectively support decision-enabled tasks in a given process instance by integrating model management support.

4.2 Model management for decision support

Model management research has been primarily driven by interests originating from management science and operations research (MS/OR) community. While a comprehensive review of the model management literature can be found elsewhere (Blanning 1993; Chang et al. 1993), it is worth noting that much of the motivation behind model management focused on finding ways for developing, storing, manipulating, controlling, and effectively utilizing models in an organization (Muhanna 1993). 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) (Geoffrion 1987). 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 (Chari 2002; Dhar and Jarke 1993; Kottemann and Dolk 1992). Model integration focuses on synthesizing models at the structural or definitional level (Basu and Blanning 1994; Dolk and Kottemann 1993; Liang and Konsynski 1993). 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.

Past research has 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 and Pick (1994) 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 networked enterprise environments.

Information systems engaged to support distributed model management activities and fulfill dynamic decision-support and problem solving requests have emerged since the past decade. DecisionNet, described by Bhargava et al. (1997), is a prototype of a web-based architecture for sharing decision models. It is based on the idea that decision models can be shared by model providers and model consumers through a centralized registry mechanism, where models can be registered and located. A data warehouse based approach for model storage has been proposed by Dolk (2000), utilizing SM approach for representing models. Huh et al. (2000) and Huh and Kim (2004) proposed a framework for distributed collaborative model management, emphasizing coordination and propagation of changes in a model base on a real-time basis. Iyer et al. (2005) recently proposed a web services architecture

for model sharing and reuse of spreadsheet models while Ezechukwu and Maros (2003) proposes an architecture for supporting distributed optimization over the Internet. Recently, Madhusudan (2007) presented a framework for distributed model management based on web services. The framework utilizes the integrated Service Planning and Execution (ISP&E) (Madhusudan and Uttamsingh 2006) for composing web services.

While these advances are significant and relevant, several research gaps can be noted, which guide the requirements of the proposed framework. The ability to share and reuse mathematical or decision models, which support underlying business processes through aligned business and decision objectives, is a core requirement. Further, the distributed nature of networked enterprises drive several other issues and design requirements (Ezechukwu and Maros 2003; Geoffrion 1987; Muhanna and Pick 1994): (1) a single model representation format, (2) representational independence of model structure and the detailed data, (3) representational independence of model structure and the model solution, (4) meta-modeling capability to support reasoning about models, (5) extensible for different modeling paradigms, and (6) accessibility of decision support resources. These requirements also emphasize the need to reason about syntactic as well as semantic knowledge embedded in models. Moreover, models are not standalone entities, but are tied to other resources such as problem specific solvers, and are often expressed in different representational formats, which present additional challenges.

5 Framework for integrated process and decision model support

In this section, the proposed framework is presented that addresses the requirements identified in Section 4. Figure 4 illustrates the proposed framework for supporting knowledge work processes. Two main components of the framework are: (1) the declarative process management platform, and (2) the model management platform. Both are elaborated in the following sections followed by a summary of implementation details.

5.1 Declarative process management platform

The declarative process management (Deokar et al. 2004) component is comprised of a process modeler, a process execution and monitoring engine, and a resources administrator.

5.1.1 The process modeler

The process modeler component is essentially an AI planner which uses the domain description, and the initial

state information to find a feasible process model (i.e., a plan) for a planning problem (i.e., a network of tasks to be completed). As depicted in the figure, the workflow modeler role is responsible for administrative management of process models through the process modeler component. In the framework, Hierarchical Task Network (HTN) action-state formalism for AI planning is used (Nau et al. 2003, 2005). Literature indicates a number of recent applications of HTN planning for modeling dynamic systems such as web services (Sirin et al. 2004). In HTN planning, the input to the planner consists of the "planning problem," a "domain description," and an "initial state" (Fig. 5). The "planning problem" is specified by an initial task network, which is a set of *tasks* (symbolic representation of activities in process modeling terminology) that need to be accomplished to meet the process objectives. Tasks may be of two types, 1) compound tasks called *methods* (which can be decomposed further), and 2) primitive tasks called *operators*. The "domain description" consists of a set of planning operators, methods, and axioms. An operator is a primitive task type describing what action the plan executor can perform, when the preconditions applicable for the task are satisfied. It also describes the effects the task may have on the state, when it is successfully executed. A method is a "recipe" of how to decompose the compound task into further subtasks when the preconditions applicable for the method are satisfied. Each method thus defines the dependencies between its subtasks (which may be compound or primitive). Various control constructs can be embedded in defining methods and operators (such as sequence, if-then-else, repeat-while, repeat-until, choice, unordered), which can guide the plan search through the state space (Sirin et al. 2004). In addition to the set of planning operators and methods, any optional information such as definition of auxiliary functions and axioms (rules) for inferring conditions not explicitly mentioned in state descriptions are also included as part of the domain description. The domain description thus represents fundamental constructs for hierarchical, modular and declarative description of activities involved in a business process. Different business processes will likely be represented through different domain descriptions. The "initial state" is a symbolic representation of the state of affairs in the process world view before planning commences.

Given the planning problem, domain description, and an initial state, the task of process modeling then is to generate a suitable set of action sequence(s) required to perform the set of tasks specified in the initial task network (i.e., the planning problem) for a particular business process instance. HTN planning is a search technique that creates plans by task decomposition, i.e., partitioning the state space effectively and efficiently using the domain knowl-
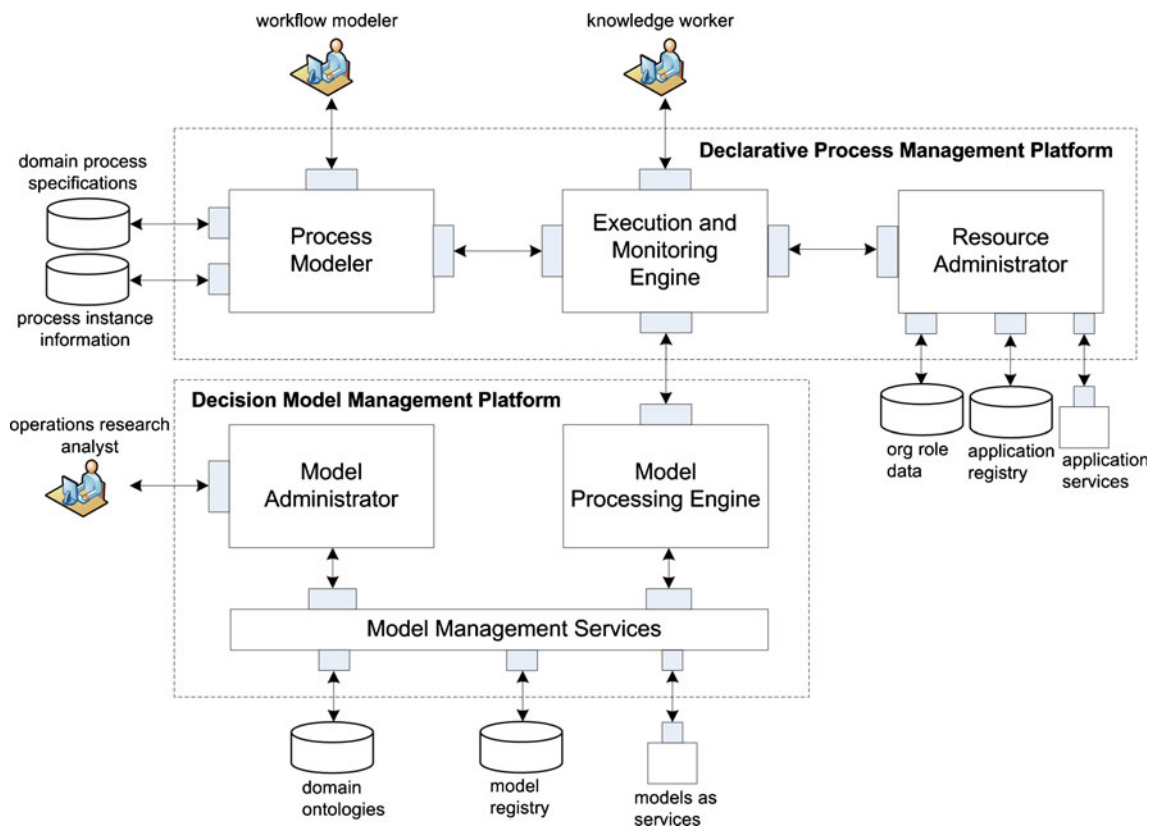
Fig. 4 Decision-enabled process management framework

edge embedded in the domain description. Planning progresses by recursively decomposing compound tasks in the initial task network into progressively smaller subtasks. This decomposition continues until the task network contains only primitive tasks (or operators). The planning algorithm itself has been detailed in Ghallab et al. (2004), and Nau et al. (2003). The planner's output is a totally-ordered sequence of operators, a plan, which when executed in a process world satisfying the initial state will achieve the process objectives.

5.1.2 The process execution and monitoring engine

The process modeler component interacts with the execution and monitoring engine in an interleaved manner. This can be explained by considering various scenarios. In cases where all knowledge is available a priori, the planning module is able to generate a complete process model ready to be executed by the execution and monitoring engine. During the execution of the process plan, anticipated or unanticipated exceptions may lead to a process state that is
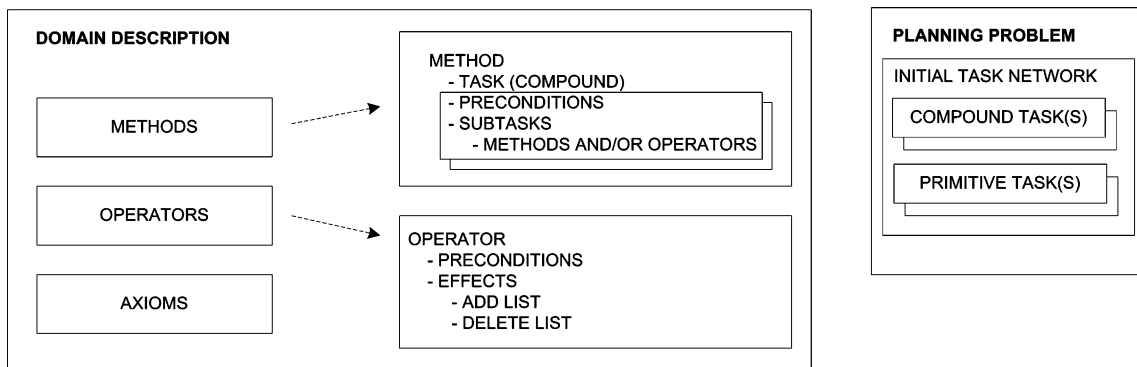

Fig. 5 Components of planning domain description and planning problem

different from the expected state (obtained from planning). The execution and monitoring engine then reactively triggers re-planning, which involves planning a new process model for the instance based on the current state of the process instance (Schuschel and Weske 2004). In many cases, complete knowledge is not available a priori because of choices that have to be made by the knowledge worker during runtime (e.g., selecting the most appropriate decision model based on available alternatives). In such cases, required information is gathered during execution, after which planning proceeds as before (Stone and Veloso 1996). Regardless of the above cases, the execution and monitoring engine also periodically checks for any changes in the domain description and the planning problem to possibly trigger re-planning. These changes will likely be a result of the agile nature of the environment where new tasks may be introduced in the domain description, or the desired goals (network of tasks to be completed) may be changed.

The execution of each of the tasks in the process is coordinated by the execution and monitoring engine. Concurrency issues are resolved during execution time by inspecting the applicable tasks in the given process state (Backstrom 1998; Madhusudan and Uttamsingh 2006). The tasks can either be executed in an automated manner through web services or facilitated by the knowledge worker, depending on the nature of the task. Knowledge worker tasks are offered to the concerned organizational role member(s) through an inbox mechanism accessible through a thin client. The knowledge worker has to accept the task from his or her inbox before commencing work on the same.

5.1.3 The resource administrator

The resource administrator component provides access to auxiliary application services that may be needed in performing several other tasks that are not necessarily decision-oriented. The application registry provides information for accessing and binding the application services to a specific task. Additionally, the resource administrator component also provides link to the organizational data needed for task allocation to appropriate knowledge workers.

5.2 Decision model management platform

The decision model management platform is shown in the lower part of the framework. It is based on the notion of conceiving decision models as a loosely coupled components delivering specific functionality in the form of web service. An analogy between models and services can be noted with respect to service-oriented principles of reuse,

abstraction, autonomy, loose coupling, statelessness, composability, and discoverability (Papazoglou 2008). This indicates a potential for significant synergistic development between model management and service-oriented technologies (Ferguson and Stockton 2005). Given that web services are self describing, self contained software applications that are accessible over the Internet, distributed resources such as decision model schemas and instances can be shared in the form of model proxy services. Additionally, executable models may be interfaced as model web services themselves. This platform is comprised of two main components: a model administrator responsible for administering and managing models and associated meta-models, and a model processing engine responsible for handling requests from the process execution and monitoring engine as detailed in the next sub-sections.

5.2.1 Model administrator

The model administrator component provides administrative access to manage models as services. Again, the model management services provide the interface for such operations (like creation, modification, storage, retrieval, deletion of models). Typically, a modeling expert such as an operations research analyst is likely to be the role member responsible for performing model administration. Model management services act as the glue between the model processing engine and decision models wrapped as services. Models (as services) are registered in a centralized registry accessible to the model management services.

Providing rich model management functionalities relies on the way models are represented. While several model representation paradigms have been proposed in the literature (Konsynski and Sprague 1986), Structured modeling (SM), originally developed by Geoffrion (1987) is noted to be a rich and widely accepted modeling paradigm for explicit representation of model structure deploying meta-modeling techniques. Supporting model representation in a web services environment, Kim (2001) and El-Gayar and Tandekar (2007) proposed XML-based representations for analytical models. Both languages are based on the SM paradigm (Geoffrion 1987) for conceiving, manipulating, and representing models at a higher level of abstraction to facilitate drawing inferences about models. Structured Modeling Markup Language (SMML) (El-Gayar and Tandekar 2007) has been used as the model representation scheme in the proposed framework.

Ontologies can be used to develop semantically rich decision models that can support intelligent reasoning and querying based on not only syntactic information, but also semantic information. These reasoning capabilities provide the necessary technological support needed to discover, interpret, compose, and execute models. Moreover, the use

of ontologies facilitates the capture of model semantics that is independent of a particular tool or application. As shown in Fig. 6, the different abstraction levels for model representation, serve as one dimension. Models from different modeling paradigms may be accommodated. Along an orthogonal dimension are the different domains. A number of domains may also be relevant in the context of a particular problem domain. For example, for a transportation model being formulated for the supply chain domain, the domain ontology may consist of key terms such as supplier, demand and customer. Additional semantics may be provided to the models using other auxiliary domain ontologies as well. For example, currency ontology as mentioned in the discussion above may be used to provide semantics to cost variables. Essentially, the problem domain ontology along with other auxiliary domain ontologies, together forms a library of relevant ontologies to provide semantics to the models. In the framework, these ontologies are represented using the Web Ontology Language (OWL).

To incorporate semantic hooks in representing models, models need to be semantically annotated with semantic concepts in semantic data models like problem domain ontologies. Also, given that decision models exist in various shapes and formats, the representation scheme needs to accommodate these differences across the spectrum. Figure 7 illustrates these ideas. On one hand, models in a binary executable format do not provide access to the model structure, and are amenable to a so-called "black box approach". Such models are wrapped as model web services. Web services are conventionally described in a procedural manner using WSDL (Web Services Description Language), which captures their functional characteristics. The SAWSDL mechanism is used to annotate models described as web services.
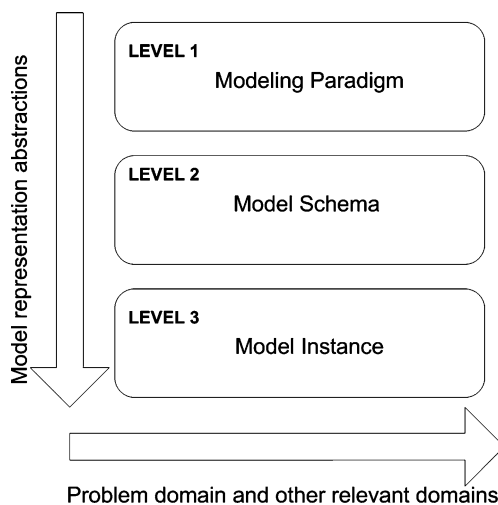

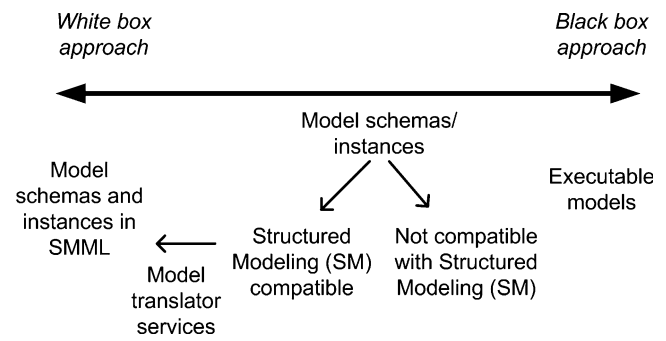Fig. 6 Model representation abstractions

Fig. 7 Representing decision models using varied approaches

On the other hand (in Fig. 6), model schemas and instances represented using SMML provide explicit access to the model schema and instance structures (and so the term "white box approach"). SMML has been extended to incorporate problem domain semantics by linking domain concepts to relevant semantic models (e.g., domain ontologies) through semantic annotations, in a manner similar to SAWSDL (Kopecký et al. 2007). This extended model representation format is referred to as SA-SMML. These models are encapsulated as model proxy web services. The model proxy web services are essentially web service wrappers for decision models and provide operations for accessing various parameters of the model, as well as solving the model through appropriate solvers. Midway along the spectrum shown in Fig. 7 lie models represented using higher level model representation languages other than SMML (e.g., LINGO, GAMS). Some of these models may be amenable to be described using SM, while others may not, depending on the kind of decision problem they represent. Models that are amenable for SM representation may be translated to SMML, and the model semantics can be captured in SA-SMML. While the white box approach provides more information in the form of the internal model structure that can support better discovery and more importantly model integration, such an approach is infeasible in situations where structured representation of a model is not possible. For the class of models that may not be described using Structured Modeling, model proxy web services have to be created using their respective model representation format. The operations provide access to various parameters of the model, and the semantics resides in SAWSDL descriptions of such services.

5.2.2 Model processing engine

Tasks involving decision models are handled by the model processing engine. This engine accesses the operations in appropriate model management services mapped to the corresponding tasks (e.g., recommend models, select models). The model management services invoke the

relevant operations on the models wrapped as services, and relay the results to the corresponding back to the execution and monitoring engine through the intermediate model processing engine.

5.3 Implementation

The key components of the framework have been prototyped using J2EE technologies. The process modeler and the execution and monitoring engine are built on JSHOP2 (Java implementation of Simple Hierarchical Ordered Planner 2—SHOP2) (Nau et al. 2003), which is an HTN planner using ordered task decomposition and constraint satisfaction as its search-control strategy. Ordered task decomposition strategy plans for tasks in the same order that the tasks will later be executed, which is particularly allows for interleaved planning and execution, as discussed earlier. The execution and monitoring engine uses Apache Tomcat for the web server functionalities. The decision models are primarily SMML models annotated with domain ontologies expresses as OWL. The SAWSDL mechanism is used to annotate models described as web services. Service-oriented technology standards have been used in linking services to various other components. Finally, Web-based forms for interacting with user roles have been developed as Java Server Pages (JSPs).

6 Illustrative case

Revisiting the motivational scenario, the framework and its different components are discussed below in the form of an illustrative case. The high level process is shown in Fig. 2. The process mainly concerns with order fulfillment for an

e-commerce company making home deliveries of consumer products such as groceries and other household items. Delivery vehicles are to be scheduled and drivers' assignments to be made on a per-order basis, given short lead times. The process elements are represented at a greater level of granularity in the planning domain description that forms one of inputs to the process modeler (AI planner). Figure 8 shows an annotated part of such a domain description.

Figure 9 depicts a Unified Modeling Language (UML) sequence diagram representing the interactions among various components of the framework. Upon execution of the optimal ordering task, the process execution engine presents the task to the knowledge worker for acceptance. Once accepted, the process execution engine request from the model processing engine to recommend model schemas belonging to a particular model type. The model processing engine in turn relies on the model management services to identify models meeting the desired criteria. Model management search and retrieval services can leverage ontologies to aid with identifying relevant models. For example, models of type "vehicle routing" may be of interest for a particular problem. The ontology may have a number of models listed under "dynamic vehicle routing" ("dynamic vehicle routing" being a subclass of "vehicle routing" in the domain ontology), then both model types match "semantically" since "vehicle routing" subsumes "dynamic vehicle routing". The process execution engine then presents the knowledge worker with the list of models. Such a list may be augmented with meta-model information pertaining to the semantics of the various elements of the model. Figure 10 provides a SA-SMML code snippet from which such semantic information may be extracted for a particular model. The knowledge worker will then select a model

```
;;--------------------------------------------------------
;; DOMAIN DESCRIPTION
;;--------------------------------------------------------
;; COMPOUND TASK
(:method (determine-optimal-order-routing ?order-id)
    ;; PRECONDITIONS - CHOICE 1
    ((order-info ?order-id ?prod-id ?quant)
     (customer-info ?cust-id ?address ?geocode-latlong)
     ...
     (model-type ?mod-type)
     (pending-model-recommendation ?order-id))
    ;; POSTCONDITIONS - DECOMPOSITION OF COMPOUND TASK
    ((!recmnd-rout-model-schemas ?mod-type)
     (!select-model-schema ?select-modid)
     (!provide-model-instance ?select-modid ?mod-data)
     (!execute-model ?sol-id ?select-modid ?mod-data))
    ;; PRECONDITIONS - CHOICE 2
    ((order-info ?order-id ?prod-id ?quant)
     (customer-info ?cust-id ?address ?geocode-latlong)
     ...
     (model-type ?mod-type)
     (pending-model-recommendation ?order-id)
     (curr-routing-plan ?exist-plan ?plan-id))
    ;; POSTCONDITIONS - DECOMPOSITION OF COMPOUND TASK
    ((!recmnd-rout-model-schemas-prior-plan ?mod-type ?exist-plan)
     (!select-model-schema ?select-modid)
     (!provide-model-instance ?select-modid ?mod-data)
     (!execute-model ?sol-id ?select-modid ?mod-data))
```

```
;;------------------------------------------------
;; DOMAIN DESCRIPTION (CONTD.)
;;------------------------------------------------
;; PRIMITIVE TASK
(:operator (!recmnd-routing-model-schemas ?mod-
type)
    ;; PRECONDITIONS
    ((model-list ?mod-type ?mod-id)
     (call RecommendedModels ?mod-id ?mod-type)
    ;; POSTCONDITIONS: DELETE LIST
    ((pending-model-recommendation ?order-id))
    ;; POSTCONDITIONS: ADD LIST
    ((models-recommended ?order-id ?mod-id))
...


;;------------------------------------------------
;; INITIAL STATE AND PLANNING PROBLEM
;;------------------------------------------------
defproblem order00102 order-management
;; INITIAL STATE
  ((order-info ORD00102(XTE001 BRD2468) (3 6))
  ...
  (pending-model-recommendation ORD00102))
;; PLANNING PROBLEM
  ((determine-optimal-order-routing ORD00102)
   (determine-optimal-driver-assignment ORD00102)
   ...))
```

Fig. 8 Process knowledge representation snippet for order processing example

Fig. 9 Execution of a task with a pre-specified model
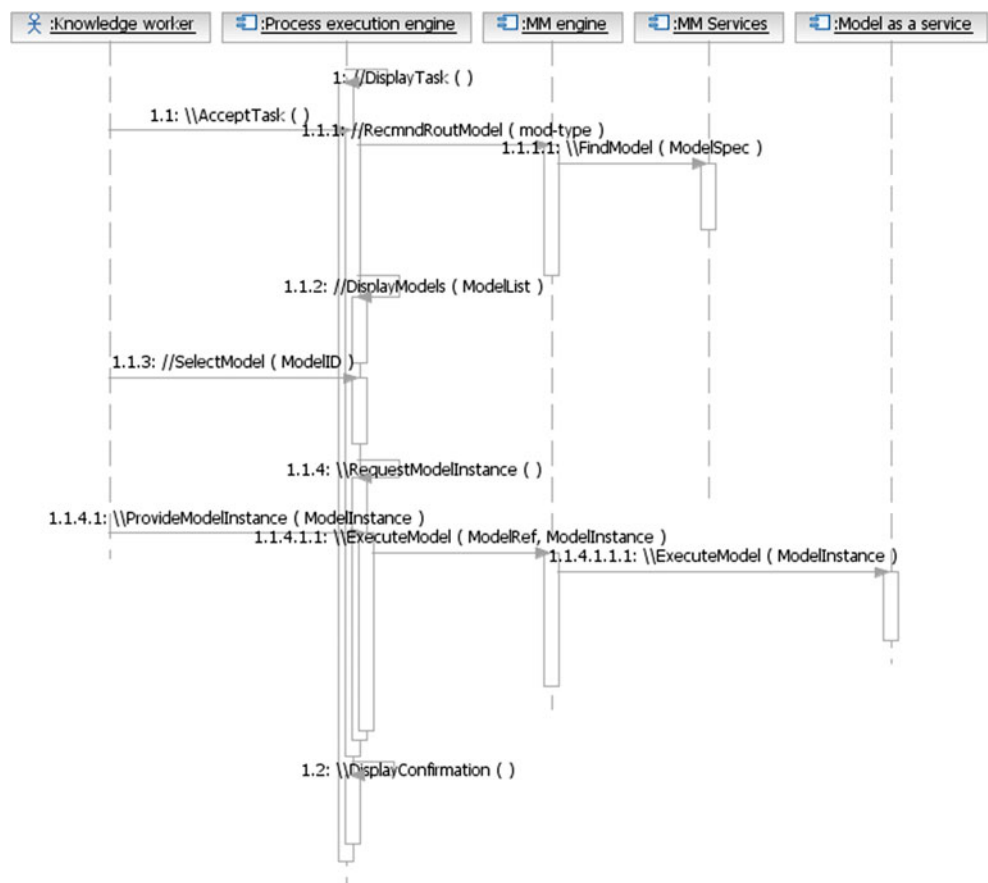
schema and provide a model instance (data) for model execution. The process execution engine forwards this information to the model processing engine for execution via the model management services.

7 Discussion

We now discuss each of the requirements identified in Section 4 from the perspective of considering whether and how they are incorporated in the proposed framework. Note that prefix P denotes process management requirements, while prefix M denotes model management requirements. (P1) Interleaved process modeling (planning) with execution and monitoring allows for designing process models

Fig. 10 Vehicle routing model representation (SA-SMML) snippet

that are customized to each process instance, given that the process planning takes the instance specific information into account while planning. (P2) The process knowledge (domain description) as well as process goals may be modified (to add, change, or remove operators or methods or change the planning problem) during execution of a process instance, thus supporting process evolution. The monitoring engine monitors any changes after execution of each task to be able to react to process evolution by triggering re-planning. (P3) Exceptions occurring during process execution are detected by the monitoring engine comparing current state with the expected state (obtained through planning) at the end of each task. Any discrepancies trigger re-planning by temporarily ignoring the operator whose execution resulted in an exception. (P4) The execution engine computes the process state after each task, and thus supports monitoring and traceability of process states. (P5) Decision-oriented tasks are handled by the execution engine through interaction with the model processing engine, which in turn invokes the appropriate model management services to facilitate completion of the task in association with relevant inputs received from the knowledge worker. (M1) SMML model representation provides for single representation format that can accommodate a variety of classes of models including LP, MIP,

and so forth. (M2) The Structured Modeling approach underlying SMML supports separation of model structure (schema) from the model instance (detailed data). This allows for invoking the same model structure with different inputs and parameters as data. (M3) The model solution generated is separately represented than the model structure, which provides the desired independence between the two. (M4) SMML capture the meta-model information structurally and syntactically, while the extended SA-SMML incorporates links to semantic concepts and thus allows capturing meta-model information semantically as well. This is indeed very useful in reasoning about models for number of model management activities such as search and retrieval of models. (M5) SMML and the underlying SM approach support a number of modeling paradigms including optimization models, and analytical models. (M6) Models are wrapped as services and registered in a registry which allows for easy accessibility and invocation of decision models. Similarly, model management services are provided for conducting higher level operations on models. In sum, the framework caters to meet the different requirements identified to support decision-enabled dynamic process management.

There have been some research and development efforts that have somewhat similar objectives. For example, (Maier et al. 2005) discuss the necessary standards, languages and tools to manage an enterprise knowledge infrastructure (EKI). In this infrastructure, and similar to the framework proposed in this paper, ontologies play an important role and representing meta-data. While the framework proposed in this paper focuses on leveraging mathematical models to support process workflows (including the sharing and reuse of models), the EKI focuses on sharing and reusing documents as an enterprise resource.

8 Conclusions and future work

This paper presents a framework that is aimed at providing decision-enabled dynamic process management support for knowledge work processes in networked enterprises such as global supply chains. The objective is to coordinate knowledge workers' processes and to provide relevant decision models needed for the efficient operation of the enterprise. The framework highlights and addresses several important issues in this context. First, the need for a seamless alignment of decision objectives, models and tools with business processes is emphasized. Next, the challenge of supporting dynamic work processes is considered. Last, but not least, the issue of providing an architecture for sharing and reusing models, that integrates with and supports the underlying process management platform is tackled.

The development of the proposed framework highlights the need to further investigate a number of issues pertaining to the future development and wide-scale adoption of the underlying technical infrastructure. These issues include:

1- The technical complexity involved in leveraging and integrating a diverse set of technologies and supporting development tools. This may be addressed through incremental development and prototyping. However, further research is needed to develop building blocks and design patterns that would simplify future development by maximizing sharing and reuse of design solutions and application program interfaces (API) for integrating various technologies.
2- The ability to integrate the proposed dynamic planning modules with existing workflow management systems. This will allow organizations to leverage existing investments in WFMSs.
3- The adoption of standards for representing decision models, e.g., SMML and SA-SMML. This is particularly relevant with the increased emphasis on inter- in addition to (intra-) organizational processes and supporting decision models.
4- The integration of the proposed framework with other proposals for developing an enterprise knowledge architecture such as (Maier et al. 2005).
5- The development, sharing, and reuse of domain ontologies. Specifically, ontology development is a time-consuming and expensive process involving a number of domain 'experts'. Any wide-scale adoption of semantic annotation of decision models will need to address issues that would facilitate the development of pertinent domain ontologies in a cost-effective manner.
6- The development and refinement of decision model search techniques that leverage semantic information in a model repository.

In conclusion, with the trend of increasingly complex and dynamic business environments, organizational processes become progressively reliant on a myriad of electronic resources distributed within and across enterprises. The introduction of technical infrastructures such as the one presented in this paper that are capable of providing the required flexibility and support for dynamic process changes as well as seamless link to decision models and tools can create opportunities for enterprises to collaborate and thrive in networked environments.

References

Backstrom, C. (1998). Computational aspects of reordering plans. *Journal of Artifical Intelligence Research, 9*, 99–137.

Bae, H., & Seo, Y. (2007). BPM-based integration of supply chain process modeling, executing and monitoring. *International Journal of Production Research, 45*(11), 2545–2566.

Basu, A., & Blanning, R. W. (1994). Model integration using metagraphs. *Information Systems Research, 5*(3), 195–218.

Basu, A., & Blanning, R. W. (1999). Metagraphs in workflow support systems. *Decision Support Systems, 25*(3), 199–208.

Bhargava, H. K., Krishnan, R., & Muller, R. (1997). Decision support on demand: emerging electronic markets for decision technologies. *Decision Support Systems, 19*, 193–214.

Bider, I., & Johannesson, P. (2005). Guest Editorial: Special Issue on Goal-oriented Business Process Modelling. *Business Process Management Journal, 11*(6), 621–623.

Blanning, R. W. (1993). Model management systems: an overview. *Decision Support Systems, 9*(1), 9–18.

Chang, A.-M., Holsapple, C. W., & Whinston, A. B. (1993). Model management issues and directions. *Decision Support Systems, 9*(1), 19–37.

Chari, K. (2002). Model composition using filter spaces. *Information Systems Research, 13*(1), 15–35.

Clark, A., & Scarf, H. (1960). Optimal policies for a multi-echelon inventory problem. *Management Science, 6*, 475–490.

Cohen, M. A., & Lee, H. L. (1998). Strategic analysis of integrated production and distribution system. *Operations Research, 36*, 216–228.

Deokar, A. V., Madhusudan, T., Briggs, R. O., & Nunamaker Jr., J. F. (2004). *A structured approach to designing interleaved workflow and groupware tasks*. Paper presented at the Americas Conference of Information Systems (AMCIS '04), New York, NY.

Dhar, V., & Jarke, M. (1993). On modeling processes. *Decision Support Systems, 9*(1), 39–49.

Dolk, D. R. (2000). Integrated model management in the data warehouse era. *European Journal of Operational Research, 122*(2), 199–218.

Dolk, D. R., & Kottemann, J. E. (1993). Model integration and a theory of models. *Decision Support Systems, 9*(1), 51–63.

Dumas, M., van der Aalst, W. M. P., & ter Hofstede, A. H. M. (Eds.). (2005). *Process Aware Information Systems: Bridging People and Software through Process Technology*. Hoboken, NJ: John Wiley & Sons, Inc.

El-Gayar, O. F., & Tandekar, K. (2007). An XML-based schema definition for model sharing and reuse in a distributed environment. *Decision Support Systems, 43*, 791–808.

Ezechukwu, O. C., & Maros, I. (2003). *OOF: Open Optimization Framework* (Departmental Technical Report No. 2003/7): Department of Computing, Imperial College London.

Ferguson, D. F., & Stockton, M. L. (2005). Service-oriented architecture: programming model and product architecture. *IBM Systems Journal, 44*(4), 753–780.

Geoffrion, A. M. (1987). An introduction to structural modeling. *Management Science, 33*(5), 547–588.

Geoffrion, A. M., & Krishnan, R. (2003a). E-business and management science: mutual impacts (part 1 of 2). *Management Science, 49*(10), 1275–1286.

Geoffrion, A. M., & Krishnan, R. (2003b). E-business and management science: mutual impacts (Part 2 of 2). *Management Science, 49*(11), 1445–1456.

Ghallab, M., Nau, D., & Traverso, P. (2004). *Automated planning: Theory and practice*. San Francisco: Kaufmann.

Huh, S. Y., & Kim, H. M. (2004). A real-time synchronization mechanism for collaborative model management. *Decision Support Systems, 37*(3), 315–330.

Huh, S. Y., Chung, Q. B., & Kim, H. M. (2000). Collaborative model management in departmental computing. *INFOR, 38*(4), 373–389.

Iyer, B., Shankaranarayanan, G., & Lenard, M. L. (2005). Model management decision environment: a Web service prototype for spreadsheet models. *Decision Support Systems, 40*(2), 283–304.

Keeney, R. L. (1994). Creativity in decision making with value-focused thinking. *Sloan Management Review, 35*(4), 33–41.

Keeney, R. L. (1996). Value-focused thinking: Identifying decision opportunities and creating alternatives. *European Journal of Operational Research, 92*(3), 537–549.

Kim, H. (2001). An XML-based modeling language for the open interchange of decision models. *Decision Support Systems, 31*(4), 429–441.

Konsynski, B., & Sprague, R. H. (1986). Future research directions in model management. *Decision Support Systems, 2*(1), 103–109.

Kopecký, J., Vitvar, T., Bournez, C., & Farrell, J. (2007). SAWSDL: Semantic Annotations for WSDL and XML Schema. *IEEE Internet Computing, 11*(6), 60–67.

Kottemann, J. E., & Dolk, D. R. (1992). Model integration and modeling languages: a process perspective. *Information Systems Research, 3*(1), 1–16.

Kumar, A., & Zhao, J. L. (1999). Dynamic routing and operational controls in workflow management systems. *Management Science, 45*(2), 253–272.

Liang, T.-P., & Konsynski, B. R. (1993). Modeling by analogy: use of analogical reasoning in model management systems. *Decision Support Systems, 9*(1), 113–125.

Madhusudan, T. (2007). A web services framework for distributed model management. *Information Systems Frontiers, 9*(1), 9–27.

Madhusudan, T., & Uttamsingh, N. (2006). A declarative approach to composing web services in dynamic environments. *Decision Support Systems, 41*(2), 325–357.

Maier, R., Hadrich, T., & Peinl, R. (2005). *Enterprise knowledge infrastructures*. Berlin: Springer.

Muhanna, W. A. (1993). An object-oriented framework for model management and DSS development. *Decision Support Systems, 9*(2), 217–229.

Muhanna, W. A., & Pick, R. A. (1994). Meta-modeling concepts and tools for model management: a systems approach. *Management Science, 40*(9), 1093–1123.

Narasimhan, R., & Mahapatra, S. (2004). Decision models in global supply chain management. *Industrial Marketing Management, 33*, 21–27.

Narasimhan, R., Talluri, S., & Mahapatra, S. (2003). *A mathematical model for evaluating multi-factor bids for agile supply base configuration* (Working paper).

Nau, D. S., Au, T.-C., Ilghami, O., Kuter, U., Munoz-Avila, H., Murdock, J. W., et al. (2005). Applications of SHOP and SHOP2. *IEEE Intelligent Systems, 20*(2), 34–41.

Nau, D. S., Au, T.-C., Ilghami, O., Kuter, U., Murdock, J. W., Wu, D., et al. (2003). SHOP2: an HTN planning system. *Journal of Artifical Intelligence Research, 20*, 379–404.

Neiger, D., & Churilov, L. (2002). *Towards decision-enabled business process modelling tools: From e-EPC to de-EPC*. Paper presented at the Proceedings of the Australasian Conference on Information Systems (ACIS '02).

Neiger, D., & Churilov, L. (2006). *Intelligent decision support through synchronized decomposition of process and objective structures*. Paper presented at the Proceedings of the 39th Hawaii International Conference on System Sciences (HICSS-39 '06), Kauai, HI.

Nilsson, A., Tolis, C., & Nellborn, C. (1999). *Perspectives on business process modelling: Understanding and changing organizations*. Berlin: Springer.

Papazoglou, M. P. (2008). *Web services: Principles and technology*. New York: Pearson-Prentice Hall.

Paraskevopoulos, D., Karakitsos, E., & Rustem, B. (1991). Robust capacity planning under uncertainty. *Management Science, 37*, 787–800.

Revelle, C. S., & Laporte, G. (1996). The plant location problem: new models and research prospects. *Operations Research, 44*, 864–874.

Rolland, C., & Prakash, N. (2000). Bridging the gap between organizational needs and ERP functionality. *Requirements Engineering, 5*, 180–193.

Scheer, A.-W. (1999). *ARIS—business process frameworks* (3rd ed.). Berlin: Springer-Verlag.

Scheer, A.-W. (2000). *ARIS—business process modelling* (3rd ed.). Berlin: Springer.

Schuschel, H., & Weske, M. (2004). Triggering replanning in an integrated workflow planning and enactment system. In B. Andras, J. Demetrovics, & G. Gottlob (Eds.), *Proceedings of the 8th East European Conference on Advances in Databases and Information Systems (ADBIS '04)* (Vol. 3255, pp. 322–335). Budapest: Springer-Verlag.

Seila, A., Ceric, V., & Tadikamalla, P. (2003). *Applied Simulation Modeling*: Duxbury Press.

Shapiro, J. F. (2007). *Modeling the supply chain* (2nd ed.). Belmont: Duxbury.

Sirin, E., Parsia, B., Wu, D., Hendler, J., & Nau, D. (2004). HTN planning for web service composition using SHOP2. *Journal of Web Semantics, 1*(4), 377–396.

Soffer, P. (2005). *On the notion of flexibility in business processes*. Paper presented at the Proceedings of CAiSE'05 Workshops, Workshop on Business Process Modeling, Design and Support (BPMDS '05).

Stone, P., & Veloso, M. (1996). User-guided interleaving of planning and execution. In M. Ghallab & A. Milani (Eds.), *New Directions in AI Planning* (pp. 103–114). Burke, VA: IOS Press.

Swaminathan, J. M., & Tayur, S. R. (2003). Models for supply chains in e-business. *Management Science, 49*(10), 1387–1406.

Talluri, S. (2002). A buyer-seller game model for selection and negotiation of purchasing bids. *European Journal of Operational Research, 143*, 171–180.

Talluri, S., & Narasimhan, R. (2003). Vendor evaluation with performance variability: a max–min approach. *European Journal of Operational Research, 146*, 543–552.

Trkman, P., & McCormack, K. (2009). Supply chain risk in turbulent environments—a conceptual model for managing supply chain network risk. *International Journal of Production Economics, 119*(2), 247–258.

van der Aalst, W. M. P., & van Hee, K. (2002). *Workflow Management: Models, Methods, and Systems*. Cambridge, MA: MIT Press.

Weber, B., Sadiq, S., & Reichert, M. (2009). Beyond rigidity—dynamic process lifecycle support: a survey on dynamic changes in process-aware information systems. *Computer Science—Research and Development, 23*(2), 47–65.

Amit V. Deokar is an Assistant Professor of Information Systems in the College of Business and Information Systems at Dakota State University. His recent research interests are in business process management, collaboration processes and technologies, decision support systems, knowledge management, and healthcare informatics. He has published several conference publications, journal articles, and book chapters in these areas. He holds a BE in Mechanical Engineering from V.J. Technological Institute, Mumbai, a MS in Industrial Engineering from the University of Arizona, and a PhD in Management Information Systems from the University of Arizona. He is a member of AIS, MWAIS, ACM, and AAAI.

Omar El-Gayar, Ph.D. is a Professor of Information Systems and Dean of Graduate Studies and Research, Dakota State University. His research interests include: decision support systems, multiple criteria decision making, and the application of decision technologies in security planning and management, healthcare, and environmental management. His inter-disciplinary educational background and training is in information technology, computer science, economics, and operations research. Dr. El-Gayar's industry experience includes working as an analyst, modeler, and programmer. His numerous publications appear in various information technology related fields. He is a member of AIS, ACM, INFORMS, and DSI.