December 2002

ARCHITECTING DECISION SUPPORT FOR THE DIGITAL ENTERPRISE: A WEB SERVICES PERSPECTIVE

P. Balasubramanian
Boston University School of Management

ARCHITECTING DECISION SUPPORT FOR THE DIGITAL ENTERPRISE: A WEB SERVICES PERSPECTIVE

P. R. Balasubramanian and G. Shankaranarayanan
Information Systems Department
Boston University School of Management
bala@acs.bu.edu gshankar@acs.bu.edu

Abstract

In the modern day digital enterprise decision support is complicated by two factors: the emergence of mobile technologies and the evolution of business webs. From a decision support perspective, these demand moving knowledge to the edge of the enterprise boundaries and creating a widespread distribution of data and model resources. They further necessitate rapid integration of decision models, secure and reliable access to the resources, and the ability to deploy these in real-time. The Virtual Business Environment (VBE) presented in this paper provides a solution that meets these demands. We have presented a framework to inform enterprises on designing their decision support architecture. The VBE is an instantiation of this architecture. Within the VBE model management is supported as a web service. The framework is defined by identifying knowledge requirements for model management and specifying the engines based on these, and then mapping the concept to the web services platform.

Introduction

Organizations collect and manage large volumes of data as a consequence of e-business and its supporting technologies. The strong push to gain business intelligence has increased the different ways the data is analyzed and the types of decision-tasks it is used for. Decision-makers are forced to become more responsive and make quicker and more dynamic decisions because of having access to resources (data, models, etc.) anywhere, anytime (including real-time). Decision-makers perform different decision-tasks using these models and share models and outcomes with several other decision-makers. This creates dynamic decision environments characterized by high frequency and a large variety of decision tasks and multiple stakeholders (model designers, model custodians, and consumers). Model management is an integral part of a digital enterprise and in such environments efficient model management demands not only providing access to the models but also providing access to the metadata (and knowledge) associated with each.

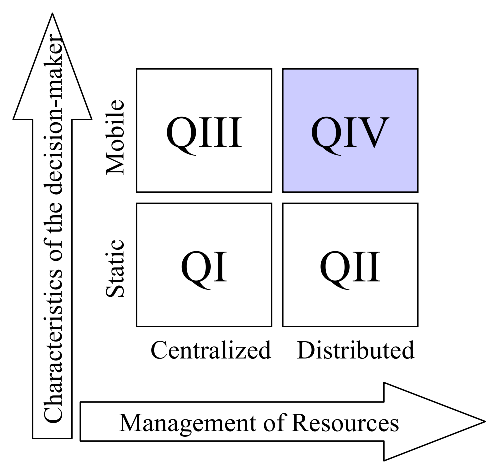

Figure 1. Decision-Support for a Digital Enterprise

The advent and widespread use of wireless technology and devices within the M-business arena promise to further complicate model management. Mobile technologies have created an environment in which the decision makers themselves are mobile. Such decision makers expect their IT groups to continue to support their workflow. Further, organizations are typically part of a business web (or extended enterprise) as a result of the alliances and relationships created for business ventures. Resources are no longer part of any one organization but span organizational boundaries. From the perspective of designing decision support architectures, mobile services impact the decision environment in two ways. First, organizations have to design architectures that support mobile decision-makers. Second, architectures should support flexible access to the distributed resources required for decision-tasks. This dichotomy results in the four quadrants as shown in figure 1. In the case of the static user and centralized resources, the major issues were around designing user interfaces and databases to store information and resulted in concepts such as decision support generators and model management systems. In QII, the reality was that resources were developed and deployed all over the enterprise. As a result, version control of models, data, and interoperability were the big challenges. This research stream resulted in concepts such as hypertext engines, Intranets/portals, agent-based search and retrieval of models, and distributed data/model management. QIII and QIV gained relevance with the advent of technologies such as 3G (3rd Generation Wireless Networks), VPN (Virtual Private Networks), extranets, ASPs (Application Service Providers) and Web Services. In QIII, an enterprise provides all the services to a mobile client. This requires secure access to enterprise resources using technologies such as VPN and extranets. With the evolution of the real-time enterprise and business webs, the QIV needs to be examined. Here resources are provided by the enterprise directly from within and indirectly via application service providers and web services.

The mobility of the decision-maker, the very widespread distribution of resources (models and data) necessary for decision-making, combined with the dynamic nature of decision-environments in organizations today demand sophisticated decision-support environments. In this paper we propose an architecture for enterprise decision support referred to as the enterprise architecture. The Virtual Business Environment (VBE) is designed based on this architecture and is hence an instantiation of this architecture. The purpose of the VBE is to ease the task of decision-makers in these complex decision environments. To facilitate convenient and mobile access to decision resources within the VBE we propose the use the of web services model. The reminder of the paper is organized as follows. The next section presents an overview of the relevant literature and further defines the scope of this paper. Section 3 presents the requirements for decision support in a digital enterprise and introduces the concept of a VBE. The instantiation of a VBE for decision support and its conceptual architecture are proposed in section 4. Conclusions and directions for further research are presented in section 5.

Relevant Literature

Model management has been recognized as one of the key components for decision support (Sprague and Carlson 1982). Research in model management can be viewed from two inter-related perspectives: model representation / integration and model storage / distribution. For representing models, the three established approaches include using the relational/semantic modeling, graph-based modeling (e.g., Structured Modeling (Geoffrion 1987)), and AI-based modeling (e.g. first-order predicate calculus, production rules (Dutta and Basu 1984)). Further, Dolk (2000) has proposed an integrated modeling environment using a data warehouse that presents a bottom-up approach for model management. In this paper, we classify model information into five different types of knowledge (or metadata) and each may be captured using different technologies as described in the next section. Since most of the model representation schemes described in literature focus on only one specific class of model information we opt for a hybrid approach for representation.

The second perspective, model storage and distribution, addresses managing models in shared / collaborative and distributed environments including the WWW. These include using object-oriented programming languages for coordinating the execution of models in distributed environments (Claire and Sharda 1990, Muhanna and Pick 1994), using “software-agents” to assist users in locating and retrieving models (Bhargava et al. 1997), and using object-models to manage and propagate changes and updates to model versions in collaborative environments (Huh et al. 2000). All of these describe mechanisms and implementation schemes for model management in shared or collaborative environments that correspond to QII in figure 1. QIII and QIV have been examined to a lesser extent. Our objective in this paper is to present the different types of data/knowledge required to clearly communicate model-information to a decision-maker, propose a conceptual architecture for capturing and disseminating this data/knowledge to mobile decision-makers (focusing on QIII in figure 1), and to describe an environment to support dynamic decision-making that we believe would characterize QIV in figure 1.

An area closely related to model management is the storage and management of data required for these models. This topic has been addressed from the perspective of distributed databases (Ozsu and Valduriez 1990, Stonebraker 1994), distributed heterogeneous databases (Hsaio and Kamel 1989), as well as managing transaction processing in such distributed repositories (Gray and Reuter 1993). The techniques described in these and other related research are very relevant for managing enterprise data resources especially so in business-webs and extended enterprises. For dynamic decision environments, data distribution (when compared with centralized data management) increases data availability, improves independence, and thus offers a more flexible solution. The rapid emergence of wireless technologies and devices for data capture (such as the radio-frequency tags and infra-red sensors) and data delivery (802.11b and wireless Internet enabled phones, handheld-computers, and PDAs) further complicate data management. In dynamic decision environments it is therefore important to deliver data to the decision-maker by recognizing the context of the decision-task so as to minimize information overload. It is also important to help the decision-maker comprehend data (e.g. results from a model) easily using techniques tailored to the individual needs of the decision-maker.

Requirements for Decision Support

There are three primary classes of stakeholders that need support for model management: the model creator (or builder), the model custodian (or analyst), and model consumer (or decision-makers) (Bala and Lenard 1998). The model creator is involved with building models by abstracting from some real life phenomenon. After understanding the domain, this person builds and maintains models using a model definition language. A custodian or analyst obtains results for decision-makers by applying models to data. Typically, in today's decision environments the custodian is also the creator and/or decision-maker except when dealing with large analytical tools such as the data warehouse in which the roles are distinct. To generate results, the analyst has to identify appropriate models, manipulate them to suit the current problem and run these models with different data sets to produce results. If the analyst can't find an appropriate model, he/she interacts with model developers to get one built. The analyst also interacts with the model consumers (decision-makers) to understand modifications required to customize the model. The last stakeholder-class, a decision-maker, is mainly interested in getting his/her questions answered. The decision-maker may, access and run the model and inspect the results or may choose to delegate the task to analysts.

Knowledge Requirements

Knowledge requirements for managing models may be classified from these three stakeholder perspectives. The first is the knowledge on how a model was created and is referred to as ***process knowledge*** (Bala and Lenard 1998). This knowledge will facilitate the building of models and their maintenance under changing environmental conditions. Model building is an iterative process during which model structures and associated assumptions are constantly revised. To reuse models, one must have the ability to modify model structures (e.g. modify parameters) for which it is necessary to have access to the formulation history in an organized manner (Dhar and Jarke 1992). This type of knowledge is obtained from the model builders and is of value to both analysts and decision-makers. The second type of knowledge is ***content knowledge.*** This captures the logical workings of a model. It is valuable when analyst wants to make some minor changes to an existing model or when a model builder wants to re-use one or more existing models to build a new one. For example, while using a profit and loss model, an analyst may want to change the formula for, say, depreciation. This would be difficult if he or she did not understand the existing formulas. Isakowitz et al. (1995) use the schema representation language (FRL) as an internal representation for spreadsheet schemas. This knowledge is obtained from builders and is updated based on feedback from analysts and decision-makers. The third is the ***operational knowledge.*** This contains information about model pragmatics. Once analysts have identified a model to use, they would like to know what data they will need to run the model, what keystrokes they should perform to get the results, how long does it take for the model to run, etc. Usually such information is contained in user manuals, if available. In addition, this layer may contain information about solution strategies, such as goal seeking, optimization, and scenario analysis. The fourth type of knowledge, ***workflow knowledge*** captures the information necessary to coordinate the use of multiple models in decision-tasks and will assist in workflow management. Workflow is the coordination of the execution of a process that is designed as a sequence of task (Kwan and Bala 1997). Workflow management systems aim to define, support and monitor the coordination of tasks in a business process. Currently, there exist several formalisms for capturing workflow knowledge: action workflow diagram (Medina-Mora et al. 1992), role interaction nets (Singh and Rein 1992), and dynamic workflow management (Kwan and Bala 1997). In dynamic decision environments, workflow knowledge would assist in "pushing" relevant models to the users as and when needed. The final knowledge layer is necessary to evaluate the quality and reliability of the model and is termed as ***evaluative knowledge.*** It includes performance metrics for evaluating the model as well as opinions of decision-makers on its usefulness and problems. It provides answers on reliability, robustness, and usefulness of the models in decision-making. Such knowledge for a model can be elicited from decision-makers who have used it or from the model builder(s). In addition, discussion groups within the organization / business web may also capture evaluative knowledge.

Web Services

We propose that model management be provided as a web service in the digital enterprise to support the widespread distribution of model resources. Besides, the web service model also provides the flexibility to rapidly develop and implement additional services by supporting component-based development of applications. There has been a great deal of hype about web services over the recent past. Microsoft, IBM, Sun, and Oracle have all been promoting their web services architectures. The definition of a web service varies considerably depending on the service provider. CNET defines web services as "self-contained, self-describing, modular applications that can be published, located, and invoked across the Web" (CNET 2001). Hagel and Brown (2001) define web services as software components that represent a business application that can talk and work with one another over the Internet via a set of open standards. Regardless of the definition, we can identify a set of key characteristics: accessibility via web, interaction via an XML interface, ability to be located via a registry, use of XML messages over standard web protocols, and support of loosely coupled connections between systems.

Several technology trends have made this new web services architecture viable. First, the ubiquity of the Internet has allowed applications to rely on services provided by others. This trend can be seen in the large number of ASP providers. Second, standards are emerging that allow web services to communicate with each other. For example, XML has emerged as a common language and financial service organizations such as RIXML have developed XML standards for their industries. Information vendors are moving away from their proprietary formats and providing data in XML format. Finally software vendors are designing and creating server software to manage web services as well as development tools (libraries) to customize and build these services. Microsoft's .NET initiative (Meyer 2001) includes several 'servers' (e.g., BizTalk Server), development tools (e.g., Visual Studio.NET) and programming languages (e.g., VB.NET and C#) that facilitate building web services. The web service model has the potential to provide significant benefits to companies and their IT organizations. In some ways, web services are a logical extension of modular design and programming. They will make information systems easier to maintain and enhance, because each self-contained service (module) will be smaller and simpler than today's large systems. IT organizations can fulfill some of their technology needs by buying services from technology vendors instead of acquiring a commercial system or developing customized applications.

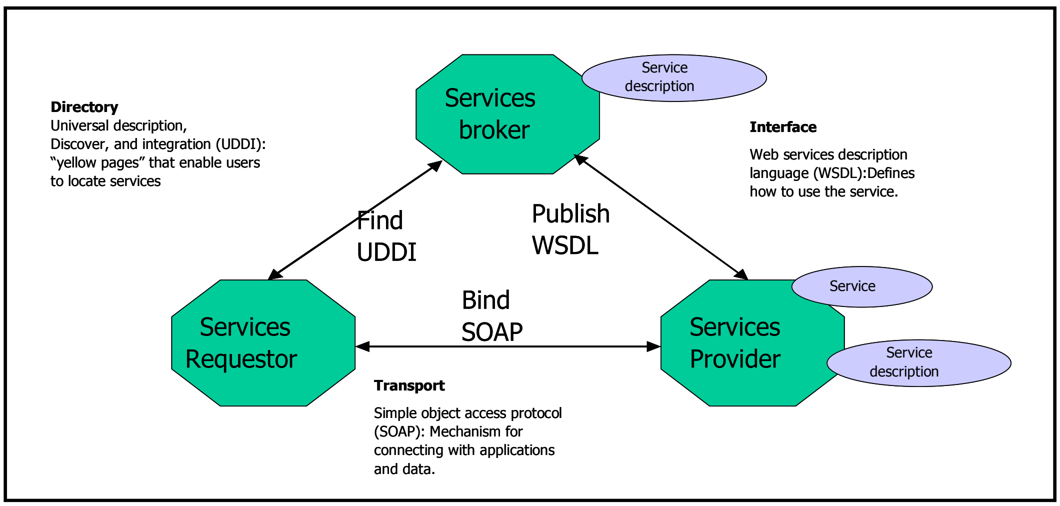

Figure 2. The Web Service Model

Virtual Business Environment (VBE)

To support decision-making in dynamic environments we propose a virtual business environment. A business environment is considered **virtual** when its instantiation at any point (spatial and temporal) is dependent on the physical surroundings and locations of the actors, decision contexts, capabilities of the devices being used, and the actors' entitlements. An enterprise can have several such environments, and its boundaries may be established on an instance-by-instance basis. The business processes (or sub processes) are defined in a virtual environment based on the needs of the decision-maker(s). The VBE would allow organizations to run different business processes concurrently and securely, while sharing resources. For example, a VBE can support concurrent availability of different business views (e.g. one for senior executives and another for project managers, both sharing data, models, and business contexts).

Conceptually, the VBE (figure 3) consists of a domain resources subsystem, a subsystem of engines, and a dialog management subsystem. The domain resources subsystem manages structured and subtly structured (non-tabular) data including expert knowledge, models and data needed for decision-making in that domain. The dialog management subsystem is responsible for interacting with the user to query information, provide inputs, and interpret the responses from the engines to create "individualized" outputs for the user. The outputs can be rendered in different ways as determined by the metaphors (captured in the user profile). The process domain typically contains software components called Business Context Engines (or simply an engine). An engine is defined as an analysis object that represents and implements a complex business capability requiring the integration of knowledge, decision-models, and data resources. Engines are shareable and reusable in disparate business contexts. The engines can be thought of as large blocks of reusable applications that instantiate complex business functions. To do so the engine needs to integrate resources, synchronize models/data, normalize outputs, dynamically allocate resources, and enforce business rules. Engines may be added to or expunged from a VBE. An engine manager maintains the library of engines, identifies appropriate engines and the necessary resources to run them. Core engines required for implementing the VBE for decision support are: (1) knowledge management engine (for managing and providing access to the model knowledge), (2) personalization engine (to customize access and display of information for decision-makers), (3) resource maintenance engine (to capture and maintain model metadata and resources), (4) delivery engine (to target delivery points and coordinate delivery of information), (5) analysis engine (to interpret results and evaluate "what-if" scenarios), and (6) a coordination engine (to coordinate the execution of multiple models and associated resources). Key design issues for the VBE, similar to research challenges tackled for distributed models (Adler 1995), include: (1) locating engines, data, expertise, and other resources distributed across the environment, (2) establishing and maintaining inter-engine communications on the network, (3) coordinating the execution of distributed engines, (4) synchronizing replicated engines or data to maintain a consistent state, (5) detecting and recovering from failures in an orderly, predictable manner, and (6) securing resources by limiting remote access to authorized users.

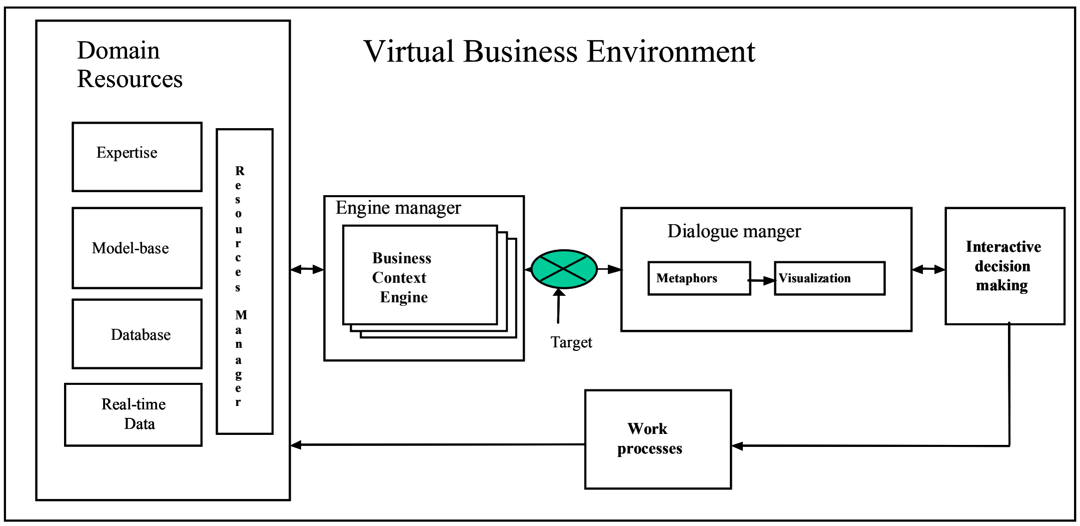

Figure 3. A Conceptual View of a Virtual Business Environment

Enterprise Architecture for Managing Model Resources

To manage model resources we need to capture the five layers of knowledge about the models and data resources. Model management must be offered as a web service to decision-makers (mobile and stationary) in the organization and in the business-web the organization is part of. Further more, the complexity of the decision-tasks in today's decision environments demands better support for the decision-maker. The creation of a VBE helps address this issue. A conceptual model of a architecture for enterprise decision-support is shown in figure 4. Engines in a VBE help define the decision-context for the decision-maker. Information about the decision-process as well as the engines required for managing the process are part of the top layer of this architecture, referred to as the ***performance (or decision) layer.*** The second layer captures the data/knowledge used in model management and is termed as the ***information layer.*** Evaluative knowledge may be captured using commercial discussion tools such as ERoom or Groove. Process knowledge may be captured using systems such as gIBIS and CADS. As shown in (Bala and Lenard 1998), the content knowledge can be represented using a variation of the Structured Model (SM) (Geoffrian 1987). Operational knowledge can also be captured using commercial discussion tools that are used for capturing process knowledge. Once a model is identified, this layer is responsible for providing the decision-makers with the information on how the execute the model(s) and how to customize it to their requirements. Further, the workflow knowledge captured in this layer supports decision-tasks that require coordinated execution of multiple models. The workflow knowledge can be captured using tools such as Role Interaction Nets (Singh and Rein 1992), Action Workflow (Medina-Mora et al. 1992), or Dynamic Workflow modeling. The bottom layer of this architecture is the ***sensory layer*** responsible for obtaining the data required to run the models. In other words, this layer includes the IT infrastructure that captures operational data in an organization. It would also include the wireless devices that "sense" data and support real-time data feeds on the movement of inventory/assets/personnel in a supply-chain or in a hospital, for example.

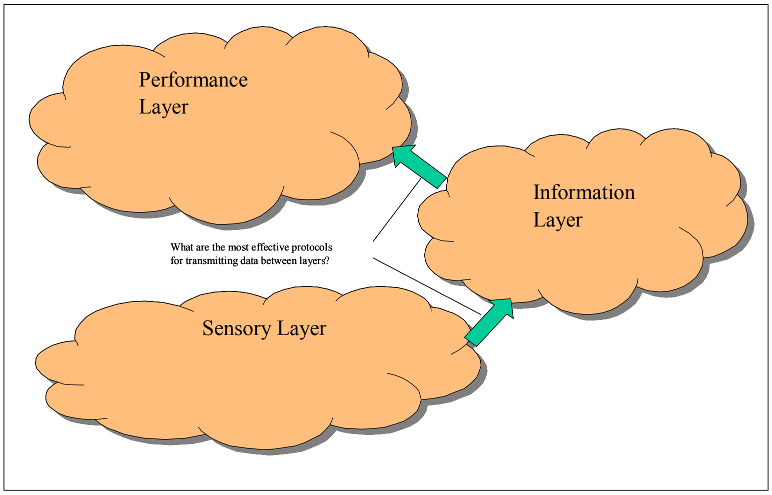

Figure 4. Enterprise Architecture

Table 1. Web Services Model (adapted from Hagel and Brown, 2001)

Application Services (Top Layer)	It consists of a diverse array of application services, from credit card processing to production scheduling, that automate particular business functions. Some applications will be proprietary to a particular company or group of companies, while others will be shared among all companies. In some cases, companies may develop their own application services and then choose to sell them on a subscription basis to other enterprises, creating new and potentially lucrative sources of revenue.
Service Grid (Middle Layer)	The service grid is the middle layer of the architecture. It provides a set of shared utilities – from security to third party auditing to billing and payment – that makes it possible to carry out mission-critical business functions and transactions over the Internet. In addition, the service grid encompasses a set of utilities, also usually supplied and managed by their parties, that facilitates the transport of messages (such as routing and filtering), the identification of available services (such as directories and brokers), and the assurance of reliability and consistency (such as monitoring and conflict resolution). In short, the service grid plays two key roles: helping Web services users and providers find and connect with one another, and creating trusted environments essential for carrying out mission-critical business activities.
Standards and Protocols (Bottom-layer)	Standards and protocols, such as SOAP and XML, allow information to be exchanged easily among different applications. These tools provide the common languages for Web services, enabling applications to connect freely to other applications and to read electronic messages from them.

The design challenges in defining a VBE described at the end of the previous section can be resolved by building engines within the VBE. We will describe the models and the knowledge about the models using UDDI (Universal Description, Discovery, and Integration). Information on how to use these models may be provided using the web services description language (WSDL from IBM). The run-time binding between the engines and the data will be implemented using the simple object access protocol (SOAP from Microsoft). A useful description of the web services model is shown in Table 1. The application services layer and the service grid layer from the web services model will reside in the manifest ***performance layer*** of the enterprise architecture. The standards and the protocol layer is the glue that binds the layers shown in the enterprise architecture (represented by the arrows between the layers in figure 4).

Discussion and Conclusion

To illustrate how the layers of modeling knowledge and the VBE might be used, consider the following example. Forecasting the earnings for the foreseeable future is a recurrent event in most organizations. In the case of Quantum Inc., this forecast was presented to every regional office during their executive committee meetings. In addition, each executive committee meeting requires customization of forecasts. Jerry, the president of the company, would generate the presentation for each office by using models that were provided by a consulting company with expertise in forecasting for the professional services market. He would then present each group with an overview and finish with a status report on performance compared with the goals that were set. He relied on his office assistant, Kathy, to provide him with the revenue projections for each visit. Kathy usually called the managers (located all over the world) for the various practices (manufacturing, financial services, etc.) to get their revenue estimates. She would then use these numbers and other assumptions to make the earnings projections. Kathy would also spend several hours with Jerry on the phone customizing presentations for each regional office.

Recently, Kathy left the company to pursue other interests. Since the quarterly meetings were fast approaching, Jerry quickly moved another employee, Julie, to her position. The first thing that Julie did was to arrange a half-day meeting with Jerry to get an understanding of the overall process. Julie wanted to know if there were any standard forms or spreadsheet models available to help her in the process. Jerry told her that Kathy had developed a spreadsheet model for the forecasting. After talking to the network administrator, Julie located the spreadsheet without much difficulty. However, she had problems in using the spreadsheet. She did not know where the assumptions were stored, or what data she had to supply, or who could provide her with the information, etc. In essence, she was looking for operational knowledge. In our framework, this knowledge would reside in the process knowledge base. For the next meeting Julie had to call the regional offices to speak to the managers for the various practices. To identify the managers she would query the workflow knowledge base.

For the next quarterly meeting, one major change was causing problems for Julie. One of the company's clients, Mtech, decided to infuse some capital into the company and a guaranteed set of projects in exchange for a 15% discount. This meant that the projected sales growth rate had to be changed to reflect the Mtech business. As Julie did not have to modify these assumptions during the previous quarter forecast, she was able to complete it. With changes to be made, she was at a loss. If she had a knowledge base containing content knowledge, she would have realized that the projected growth rate was 10% for the next three quarters and 20% for the following two. Knowing this, she could confidently add the guaranteed revenues from Mtech to the model estimate the new net earnings. Furthermore, the strategic consulting firm went out of business. However, there were other firms that provided similar forecasting model services. Unfortunately, to use those services Julie had to get someone from IT to write interfaces to integrate with the new company and their service offerings.

We presented a model for capturing information and knowledge about organizational resources and a way to provide access to those resources within a VBE. Such an architecture would help Julie search for information about the assumptions implicit in the spreadsheet Kathy had used. Moreover, the web services architecture would allow her to unplug the previous forecasting model and plug in the new model without support from IT. The evaluative knowledge about the new forecasting model would help Jerry understand the reliability of the estimates. The content knowledge about the spreadsheet would allow Julie to make the changes mandated by the influx of funding from Mtech. The workflow knowledge would enable her to contact the right person(s). Finally, the VBE provides Jerry the ability to connect to the various distributed resources while on the road or within the regional offices allowing him to dynamically customize these presentations by executing the different models with appropriate data.

In this paper we have presented a conceptual architecture for a VBE designed to support easy and flexible access to resources using a web service model for managing models and other resources. The motivation for this architecture stems from the need to support the decision-maker in the face of widely distributed resources and the proliferation of wireless technology and devices. The web services model provides the underlying technology and platform for decision support within the VBE. The same web services technology allows the enterprise to plug and play with business webs. Currently, we are building a facility that we call the Real Vision Laboratory (RVL) based on the architecture described. The RVL will serve as a platform to support and build multiple Virtual Business Environments. The engines to support model management and decision-making will be incorporated into the VBEs. We are also implementing a framework that permits decision-makers evaluate and visualize the quality of data/models in this VBE. We are further studying the impact of visual metaphors on decision-making. These are expected to provide the necessary insights and define requirements for implementing the RVL.

References

Adler, R. M. "Distributed Coordination Models for Client/Server Computing," *IEEE Computer*, vol. 28, pp. 14-22, 1995

Balasubramanian, P. R. and Lenard, M. L. "Structuring Modeling Knowledge for Collaborative Environments," presented at 31st Hawaii International Conference on Systems Sciences, Hawaii, 1998.

Bhargava, H. K., Krishnan, R., Roehrig, S., Casey, M., Kaplan, D. and Muller, R. "Model Management in Electronic Markets for Decision Technologies: A Software Agent Approach," presented at 30th Hawaii International Conference on System Sciences, Hawaii, 1997.

Claire, B. L. and Sharda, R. "An Object-oriented Architecture for Decision Support Systems," presented at International Society for DSS Conference, 1990.

CNET - http://enterprise.cnet.com/enterprise/0-9560-7-6205175.html, 2001

Dhar, V. and Jarke, M. "Conceptual Modeling and Change Propagation," in *Information Systems and Decision Process*, E. Stohr and B. Konsynski, Eds. IEEE Computer Society Press, 1992, pp. 217-230

Dolk, D. R. "Integrated Model Management in Data Warehouse Era," *European Journal of Operational Research*, vol. 122, pp. 199-218, 2000

Dutta, A and Basu, A. "An Artificial Intelligence Approach to Model Management in Decision Support Systems," *IEEE Computer*, vol. 17, pp. 89--97, 1984

Geoffrion, A.M. "An Introduction to Structured Modeling," *Management Sciences*, vol. 33, pp. 547-588, 1987

Gray, J. and Reuter, A. *Transaction Processing: Concepts and Techniques*. San Mateo, CA: Morgan Kaufmann, 1993

Hsiao, D. K. and Kamel, M. N. "Heterogeneous Databases: Proliferations, Issues, and Solutions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 1, pp. 45--62, 1989

Huh, S-Y, Chung, Q. B., and Kim, H-M. "Collaborative Model Management in Departmental Computing," *Informs*, vol. 38, pp. 373--389, 2000

Hagel, J. and Brown, J. S. "Your Next IT Strategy," *Harvard Business Review*, pp. 105--113, 2001

Isakowitz, T., Shocken, S., and Lucas, H. "Toward a Logical/Physical Theory of Spreadsheet Modeling.," *ACM Transactions on Office Information Systems*, vol. 13, pp. 1-37, 1995

Kwan, M. M. and Balasubramanian, P. R. "Dynamic Workflow Management: A Framework for Modeling Workflows," presented at Hawaiian International Conference on System Sciences (HICSS-30), Hawaii, 1997

Medina-Mora, R., Winograd, T., Flores, R., and Flores, F. "The Action Workflow Approach to Workflow Management Technology," presented at 4th Conference on CSCW, New York, 1992

Meyer, B. *Bertrand Meyer's .NET Training Course Workbook*. New Jersey: Prentice Hall, 2001.

Muhanna, W. and Pick, R. A. "Meta-modeling Concepts and Tools for Model Management: A Systems Approach," *Management Sciences*, vol. 40, pp. 1093--1123, 1994

Ozsu, M. T., Valduriez, P. *Principles of Distributed Database Systems*, 1st ed. New Jersey: Englewood Cliffs, 1991.

Singh, B. and Rein, G. L. "Role Interaction Nets: A Process Description Formalism," MCC Tech, Austin CT-083-92, 1992.

Sprague, R. H. and Carlson, E. D. *Building Effective Decision Support Systems*. Englewood Cliffs: Prentice-Hall, 1982.

Stonebraker, M. "Readings in Database Systems,", 2nd ed. San Francisco: Morgan Kaufmann, 1994.