ENTERPRISE DECISION SUPPORT SYSTEMS INTEGRATION: AN OBJECT REQUEST BROKER APPROACH

Benjamin Khoo, New York Institute of Technology
Guisseppi Forgionne, University of Maryland, Baltimore County
Peter Harris, New York Institute of Technology

ABSTRACT

Modern enterprises face a highly dynamic environment that requires management to integrate distributed and disparate knowledge for effective decision-making. The advent of the Internet has resulted in a trend toward network centric computing. As a result, more of the computing work is delegated to the computer and the underlying systems. For many years, researchers have been working towards the development of shareable and re-usable problem-solving components to support decision-making. Internetworking technologies provide a new means of sharing decision support functionalities and delivering decision support capabilities. Corporate decision makers are not willing to cede control of corporate data and models to internet-based DSS, as a result enterprise-wide knowledge portals (corporate intranet-based DSS) has been developed to overcome this shortcoming. The Common Object Request Broker Architecture (CORBA) is one such integration approach. This paper describes how this approach can improve the enterprise decision support systems.

JEL: O3; L8; M1

KEYWORDS: Decision Support System, Enterprise System Integration

INTRODUCTION

Computer based systems developed to support decision making are called decision support systems (DSS) (Bhargava and Power, 2002; Keen and Scott Morton, 1978). DSS often consists of models and algorithms. Organizational knowledge (or corporate memory) is the informational, technical, historical and domain expertise that an organization possesses. (Rabarijaona et al., 2000) have defined corporate memory:

> *A corporate memory is an explicit, disembodied and persistent representation of knowledge and information in an organization, in order to facilitate their access and reuse by members of the organization, for their tasks.*

This knowledge is often captured in different departments within the organization, very often in the form of data stored in databases, technical documentation, and processes, models, tools and/or algorithms implemented in computerized systems. In addition, the skills possessed by individuals can be captured in expert systems or knowledge based systems developed within the organization.

In the mid-twentieth century, departments in an organization were able to function as close-knit units. However, with the technological explosion of the Internet, organizations became more high tech, efficient, and more complex as technology hid most day-to-day operations from human users. Late in the twentieth century, organizations were so large and complex that often one department did not know what the other related departments were doing. Currently, organizations operate in a highly dynamic environment, requiring management to respond speedily and flexibly to external changes (Turban, et al, 2007). Organizations now must compensate for the loss of visibility of overall organizational functions due to the increased complexity and advancement in technology if they are to remain competitive (Applegate, et al, 2005; Eom, 2002).

During this same time, many advances have been made in distributed heterogeneous technologies, in the development of multi-platform programming languages and in the improvements in hardware including processing speeds and storage technology. The rapid adoption by businesses of the Internet, Intranet and Extranet and the advent of new programming languages such as Java ("write once, run anywhere") and JavaScript, combined with the client-server architecture adopted by many enterprises, has opened new opportunities for information systems researchers to develop distributed, network centric systems. These advances coupled with advances in information technology (IT), has opened up more possibilities for information systems and DSS researchers to investigate and find a more efficient and enhanced way to aid the decision-making or the problem solving process. Research in information systems has been driven by these technological trends (Bhargava, Power and Sun, 2007). As a result, the trend is to delegate to the computer and the underlying systems more of the computing work. In this new model, the content, communication, and computing converge on the network resulting in the network becoming the COMPUTER. Advances in these technologies have contributed to the development of advanced IS.

For many years, researchers have been working towards the development of shareable and re-usable problem-solving components to support decision-making (Chari, 2003; Gachet, 2002; Carlsson & Turban, 2002; Krishnan & Chari, 2000). Industrial researchers, from the mid-1990s to the 2000s (Vinoski, 2003; Whitehead, 2002; Vinoski, 2002; Hummingbird Whitepaper, 2001; Firestone, 1999; Dabke, 1999; Clarke, Stikeleather & Fingar, 1996; Vinoski, 1997; Firestone, 1997), had consistently been expounding the need to integrate these distributed and disparate knowledge systems as a possible solution to support a quick response mechanism in an organization. The aim is to build flexible component-based systems that are adaptable. However, this concept has not been implemented successfully on a large-scale in cross-platform components due to a lack of standards and the enormous costs involved with the implementation.

This paper offers a systems concept that can help deliver an integrated decision support system (IDSS) within an enterprise. We begin with a literature review. The next section discusses current enterprise issues; this is followed next by a description of the integrated framework. We then introduce the IDSS from the distributed perspective. Finally, we discuss the CORBA delivery mechanism to implement the IDSS. We conclude this paper with a discussion of the applied research contribution of this paper.

LITERATURE REVIEW

The DSS domain has had an illustrious history, beginning formally in the 1960s (Keen and Morton, 1978). There have been many papers written about the general state of the art in DSS and its associated technologies, for example, Burstein and Holsapple (2008), Bhargava, Power and Sun (2007), Power and Sharda (2007), JP Shim, et al (2002), Decision Support Systems Journal, volume 33 (2002) and volume 43 (2007), Powell (2001), Power (1999), Power and Karpathi (1998) and Power (1997) amongst others. This section will focus on a review of the literature related to the research in DSS and the enterprise system domain. Research and development in this area has been slow over the past decades.

Management is an integral function of an organization. A principal component of management is decision making, which is a complex process (Keen & Scott Morton, 1978, Turban, et al, 2001). Herbert Simon's model of decision-making involves four phases - intelligence, design, choice and implementation (Simon, 1977). Traditional problem solving involves recognition that a problem exists, identifying possible causes, developing alternative solutions, choosing among alternative causes of actions and carrying out the chosen action. Most organizational problems can be categorized into: structured, semi-structured or ill-structured problems. A structured problem can be solved by applying decision rules or routines through a programmed system. Ill-structured problems require an intuitive-type response; it cannot be solved by programmed systems. Semi-structured problems fall somewhere in between the former and the latter (Turban, et al, 2007).

Decision Support Systems and Technologies

The practice of management, which includes a large amount of decision-making, is a complex process. Managers will need to use the support of the art, science and technology to be more efficient. Past research in decision support systems have highlighted the importance of technology to the decision making process (Zhang and Goddard, 2007; Bhargava, Power and Sun, 2007; Forgionne, 1991a). Information technology is the conglomeration of a few diverse research domains, each of which has contributed to the advancement of Information Systems (IS) research in general and DSS in particular(Burstein and Holsapple, 2008).

The technology of management (Forgionne, 1991b) can be used to assist individual and groups to work with the management process. It is the vehicle that is used to integrate the art and science of the management process. The technology of management, when used in an intelligent and creative way will help managers to increase their knowledge in an independent manner and thus, become more self-educated and a self-reliant decision-maker. A framework that can be used to provide a complete and integrated decision making support is the decision technology systems (Forgionne, 1991b).

Modern corporations and their strategic business units continue to get leaner - hierarchical organization structures are being replaced by leaner, flatter organization structures. The objective is to create business entities that are leaner, more flexible and more responsive to the dynamics in the business environment (Carlsson & Walden, 2000). As the business context gets more complex for planning and decision-making, there is a great need for support systems to be driven by advanced IT. In addition, with the increased dynamics of the business context, there will be a need for more accurate, more diverse and current information. This also drives the need for the use of advanced IT to support planning, decision-making, operations and management.

Of all the technologies that influenced the development of DSS, internetworking or web technologies is arguably the most prevalent one today. Web technologies provide a new means of sharing decision support functionalities and delivering decision support capabilities. (Power, 2000a) and (Power, 2000b) suggested some frameworks for organizing DSS on the web. Web technologies have also made it possible to implement DSS using the other approaches, apart from the data-driven and model-driven approaches, like the communication-driven, knowledge-driven and document-driven approaches. The communication-driven approach utilizes the web communication technologies to assist decision makers who might be at different locations, at different time to collaborate and resolve problems. The knowledge-driven approach utilizes the web technologies to recommend and deliver recommended actions to a broader spectrum of decision makers. The document-driven approach utilizes web technologies to integrate the storage, retrieval and processing of different types of documents for decision makers to read and analyze (Bhargava and Power, 2002).

Developers have started to develop web enabled DSS as services which can be accessed from anywhere through an Internet connection. The services can combine multiple components from different sources to deliver application solutions. Papers by (Cohen et al., 2001), (Shim, et al, 2002) and (Bhargava, Power and Sun, 2007) described some of the services that were enabled by web technologies. (Bhargava and Krishnan, 1998) classified web technologies into three main categories: enabling server-side computation (e.g. Java Server Pages, Active Server Pages, Java applications, etc.), enabling client-side computation (e.g. Java applets, client-side scripting languages, etc.) and enabling a distributed implementation and deployment of DSS components (e.g. CORBA, Java RMI, Java Beans, etc.) Web technologies are applicable not only on the Internet but it is currently playing an increasingly prominent role in enterprise intranets. The technologies that can be utilized to enable intranet-based DSS have not been fully explored.

Services Oriented Architecture

Over the past two decades, enterprise IT organizations have tried to integrate enterprise systems through the use of IT standards but often with limited success. The latest hype is the Services Oriented Architecture (SOA).

The Gartner Hype Cycle characterizes the typical progression of an emerging technology from over enthusiasm through a period of disillusionment to an eventual understanding of the technology's relevance and role in a market or domain based on human attitudes to technology. The first phase of the hype curve is driven by vain hype — mainly by the media, which speculates on the technology's prospects. The second part of the hype curve primarily is driven by performance gains and adoption growth (Linden and Fenn, 2003).

From the Gartner Hype Cycle (McCoy, et al., 2003), the Common Object Request Broker Architecture (CORBA) is the precursor of SOA on which web services is based. The status of the current integration and middleware technologies (Fenn, et al, 2008) are as follows:

1. On the rise (between Technology Trigger and Peak of Inflated Expectations) are Erasable Paper Printing Systems, Context Delivery Architecture, Behavioral Economics, Mobile Robots, Augmented Reality, Surface Computers, Cloud Computing and 3-D Printing.
2. At the Peak (Peak of Inflated Expectations) are Micro blogging, Green IT, Social Computing Platforms and Video Telepresence.
3. Sliding Into the Trough (Trough of Disillusionment) are Solid-State Drives, Public Virtual Worlds, Web 2.0, Service-Oriented Business Applications, Virtual Assistants, RFID (Case/Pallet), Corporate Blogging, Idea Management, Social Network Analysis, Wikis and Electronic Paper.
4. Climbing the Slope (Slope of Enlightenment) are Tablet PC, SOA and Location-Aware Applications.
5. Entering the Plateau (Plateau of Productivity) are Basic Web Services.

SOA provides the blueprint that supports Web Services. Web Services depends on the availability of implemented standards. When the standards are available and with solid vendor support, enterprises will be able to enable seamless inter-connectivity between clients and servers over the Internet. In contrast, CORBA is an object oriented distributed architecture based on the notion of objects invoking other objects by means of formal protocol and well-defined interfaces.

CORBA is a mature technology while SOA is still being experimented to provide better web services. SOA is actually a derivative of CORBA. SOA is a blueprint that provides a technology-independent, high-level concept that focuses on dicing and slicing of the enterprise system in a way that allows the creation of independent components that are related to business functionality while also promoting the global integration of these components. Thus, SOA is a collection of services, which communicate via simple message passing or through the coordination of two or more services. The earliest SOA made use of object request brokers based on the CORBA specification. A service is a function or method that is self-contained and is independent of other functions. Web services are an implementation of SOA. Web services are a collection of technologies such as Extensible Markup Language (XML) which is a standard documentation format for passing data, Simple Object Access Protocol (SOAP) which is responsible for encoding XML messages so that they can be received and understood by any operating system over any type of network protocol, Web Services Description Language (WSDL) which is the XML-based language that businesses use to describe their services in the UDDI and Universal Description, Discovery and Integration (UDDI) which is an XML-based directory that allows businesses to list themselves, find each other and collaborate using Web services (Bih, 2006). Even though the operational concept of Web Services and CORBA are very similar, yet there are fundamental differences.

Table 1: Comparison between CORBA and Web Services (de Jong, 2002)

Characteristic	CORBA	Web Services
Protocol	Internet Inter-ORB Protocol (IIOP), General Inter-ORB Protocol (GIOP)	Hyper Text Transfer Protocol (HTTP), SOAP, XML Schema
Location Identifiers	Interoperable Object References (IOR), Unique Resource Locator (URL)	Unique Resource Locator (URL)
Interface Definition	Interface Definition Language (IDL)	WSDL
Naming, Directory	Naming Service, Interface Repository, Trader Service	UDDI

This table compares the characteristics of CORBA versus that of Web Services. The characteristics are not a one-to-one comparison, for example, UDDI does not correspond exactly to the Naming Service and Trader Service but is somewhere in between them.

As stated earlier, Web Services is actually a remake of the CORBA wheel. Much hype is revolving around Web Services as industry considers it far more useful and intellectually attractive to discuss this "new" technology. Publishers have to sell publications and stable technologies such as CORBA is not attractive to the consumer. As Web Services are still evolving, there are some issues or disadvantages when compared to CORBA (de Jong, 2002; Gray, 2003):

1. Constructing a SOAP message for Web Services is more complex whereas CORBA is closer to the coding style of the programming language.
2. SOAP interface XML is more difficult to read than CORBA's IDL.
3. Web Services uses the HTTP transport that can result in the saturation of port 80 whereas CORBA uses optimized connection-oriented communication protocols.
4. The use of HTTP and XML text documents in Web Services increases inter-operability but can result in significant increase in run-time cost compared to CORBA.
5. Web Services are designed to operate through the Internet but the issue of security has not been adequately addressed whereas CORBA is more applicable to an intranet and is inherently more easily secured.

ENTERPRISE ISSUES

Web technologies are currently playing an increasingly prominent role in enterprise intranets. Corporate decision makers are not willing to cede control of corporate data and models to internet-based DSS, as a result enterprise-wide knowledge portals (corporate intranet-based DSS) has been developed to overcome this shortcoming. The technologies that can be utilized to enable intranet-based DSS have not been fully explored. Further research in this direction is needed to fully extend the capabilities of corporate intranet-based DSS.

Since the last century, organizations have been motivated to develop the rules of work design based on a model of decentralization (specialization of labor) and economies of scale derived from the Industrial Revolution. This motivation resulted in the propagation of departments within organizations, each specializing in a specific work process. Over time, each of these departments develops some form of computerized models that help to enhance its own productivity. These systems are known as "islands of automation" due to the distributed isolated nature of these systems. In a typical enterprise, there are "islands of automation" distributed throughout the enterprise. These systems are often computer-based systems or tools that had been developed or COTS (commercial off the shelf) systems customized for specific applications. Each of these systems, which contain models, can be perceived as a problem-solving component.

These distributed components support the enterprise's mission in different ways at various times. These distributed components are developed, based on specific departmental functionality. For example, in a production department, there are computer-based systems and models for production scheduling, capacity planning and line loading, material requisitions and production yield tracking. While in a sales department, there will be systems for sales forecasting, generating marketing strategy and plans, customer representative and customer data, and product pricing. These components or models are much departmentalized and are distributed.

The disparity and distribution of knowledge (computerized models, decision support systems, and other expertise) has made it difficult to capture, access, and reuse enterprise knowledge, or corporate memory (Rabarijaona et al., 2000). Such ineffective knowledge management greatly diminishes the enterprise's ability to respond rapidly to external changes. The distributed and disparate nature of these knowledge systems needs to be coherently integrated to "promote knowledge growth, promote knowledge communication and in general preserve knowledge within an organization" (Steels, 1993). Integrated systems can be developed to overcome the distributed nature of existing DSS or models with these advanced technologies (March and Hevner, 2007; Zhang and Goddard, 2007; JP Shim, et al, 2002).

APPROACH FOR INTEGRATED DECISION SUPPORT

The need for information has resulted in the development of technologies that integrate information from multiple, disparate operational systems to support problem solving. In addition, there has been widespread adoption of scalable, open system-based information technology. These factors put tremendous pressure on the problem solving process. At the source side, enterprises are accumulating data at an exponentially increasing rate. The "problem space" now becomes more complex with more data to consider for a specific problem. At the sink side, decision-makers now require more knowledge to be effective. This places tremendous stress on the traditional decision support systems.

Currently, research is being conducted to bring developments in Knowledge Management (KM) to deliver the entire organization's expertise to bear on problems anywhere and anytime (Firestone, 1999). The organization-integrated system is a system that has the collective knowledge and/or capabilities of different models, which can be of different types or similar types (for example, quantitative models, soft computing models like neural networks and simulation models) developed for different types of applications (for example, queuing, line balancing and inventory control) (Shim, 2002; Smironov, 2006). An enterprise-integrated system will also provide an enterprise secure access to its corporate memory.

IDSS can be developed to overcome the distributed nature of existing DSS or models with advanced technologies. The integration approach should: (a) be able to integrate software on diverse platforms, (b) be proven and use mature technology (or has strong industrial support), (c) provide intelligent and dynamic model access, (d) enable the models to have location transparency, (e) enable the user interface to adapt to both knowledgeable and novice users (by providing guidance), and (f) enable the knowledge gained over time to be leveraged to solve similar problems (Bhargava, et al, 2007; Gachet, 2003; Gachet, 2002).

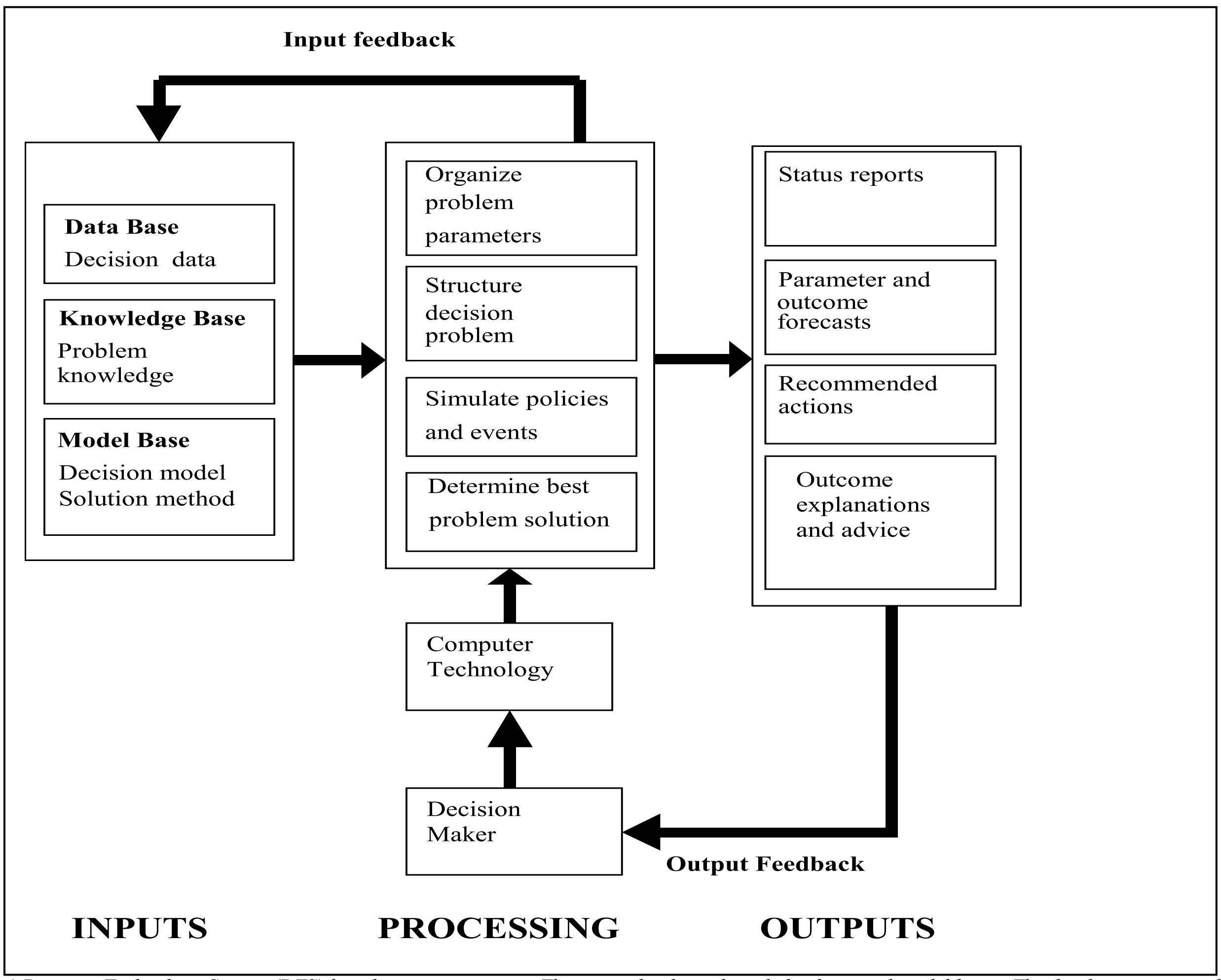
The integration of component re-use is now possible with the introduction of the Common Object Request Broker Architecture (CORBA) (and currently, CORBA 3.1) specifications. When multiple entities can collaborate to perform the reasoning process, the process is usually more efficient and the solution usually is a more optimal one (Gachet, 2002; Smirnov, 2006). The CORBA specification (OMG, 2004) describes how software components can inter-operate across networks, languages and platforms. In addition, since the release of the specifications, a number of software companies have aggressively developed products that conform to or incorporate the standard, including Iona, Inprise, Sun Microsystems, Microsoft and other vendors. CORBA can be used to deliver the integrated system by integrating the various problem-solving components (which are modular DSSs).

A framework was developed for this integrated system. A framework in this paper refers to a conceptual structure for evolving a corporate infrastructure resource that can be shared by many users and services. To be effective, a framework must satisfy three, often conflicting, requirements (Porter, 1990): (a) it should provide as much vendor-independence as practical; (b) it should be capable of rationalizing the frequent multi-vendor, multi-technology chaos of incompatible elements; and (c) it should incorporate the standards being adopted or likely to be adopted by leading vendors and electronic trading partners. The backbone of the framework is provided by the CORBA services (Schmidt & Vinoski, 2004; Krishna, et al, 2003; Brose et al., 2001).

INTEGRATED SYSTEM FRAMEWORK

An integrated system within an enterprise is hereby proposed to assist an enterprise in leveraging corporate memory. Figure 1 presents the framework of the proposed system.

Figure 1: Integrated System Framework (Forgionne, 1991b)



A Decision Technology System (DTS) has three major inputs. There is a database, knowledge base and model base. The database contains the data directly relevant to the decision problem, including the values for the uncontrollable events, decision alternatives and decision criteria. The knowledge base holds problem knowledge, such as formulas for converting available data into the problem's parameters, guidance for selecting decision alternatives and problem relationships, or advice in interpreting possible outcomes. Such knowledge includes the corporate memory relevant to the decision problem. The model base is a repository for the formal (tabular, graphic, conceptual or mathematical) models of the decision problem and the methodology for developing results (simulations and solutions) from the formal models.

Decision-makers utilize computer technology (hardware and software) to process the inputs into problem-relevant outputs. Processing will involve: (a) organizing problem parameters – accessing the data base, extracting the decision data, and organizing the information in the form needed by the solution model and methodology; (b) structuring the decision problem – accessing the model base, retrieving the appropriate decision model, and operationalizing (attaching organized parameters to) the decision model; (c) simulating policies and events – using the operationalized decision model to perform the computations needed to simulate outcomes from user-specified alternatives and then identifying the alternative (or alternatives) that best meets the decision criterion (or criteria) among those tested; (d) finding the best problem solution – accessing the model base, retrieving the appropriate solution method, and using the retrieved method to systematically determine the alternative (or alternatives), among all possible alternatives, that best meets the decision criterion (or criteria). The system can use problem ideas, concepts and knowledge drawn from the knowledge base to assist users in performing these processing tasks.

Processing will generate status reports, forecasts, recommendations and explanations. The status reports will identify relevant uncontrollable events, decision alternatives and decision criteria and show the current values for these problem elements. Forecasts will report the events and alternatives specified in the simulations and the resulting projected values of the decision criteria. The recommendations will suggest the values for the decision alternatives that best meet the decision criteria, and the corresponding criteria values, under current and forecasted values for the uncontrollable events. Explanations will justify the recommendations and offer advice on further processing. Such advice may include suggestions on interpreting the output and guidance for examining additional scenarios.

The system provides both input and output feedback to the user. Input feedback from the processing provides additional data, knowledge and models that may be useful for future decision-making. Output feedback (which can include outcomes, cognitive information, task models and what-if, goal-seeking and other types of sensitivity analyses) is used to extend or modify the original analyses and evaluations.

CORBA DELIVERY MECHANISM

The notion of distributed decision support systems each with its own capability collaborating to solve a problem was unheard of a decade ago. Today, with the advancement of technology and software, it is possible to develop such an IDSS. Some mechanisms are required for such a scenario to work:

1. Discovery mechanism: where a component can know the existence of other components and their capabilities, (location transparency can be inherent in the system),
2. Communication mechanism: where components can communicate with one another,
3. Interfacing mechanism: where components can interface with each other or with database systems,
4. Reasoning mechanism: where an inference engine can reason about the capabilities of other components so that the correct component can be invoked to solve the problem at hand.

CORBA is the vehicle used to deliver the IDSS by integrating the various DSS modules. When multiple entities can collaborate to perform the reasoning process, the process is usually more efficient and the solution usually is more optimal. The CORBA specification describes how software components can inter-operate across networks, languages and platforms (Vinoski & Lea, 2003). CORBA is a blueprint that specifies how distributed components that run on different platforms can be harnessed to collaboratively aid decision-making. CORBA is not an application package like an Enterprise Resource Planning (ERP) package. It is a toolk that needs to be programmed and configured to suit each enterprise. Computer hardware is also required to set-up the Naming Service, Interface Repository, etc. Some commercial CORBA software packages include Orbix, Artix (formerly from IONA Technologies but is now owned by Progress Software). There are also freeware versions from the Java Community. The costs for the implementation of CORBA vary from one enterprise to another, as each enterprise is unique.

CORBA provides the services to enable enterprise-wide systems integration and in so doing, provide the framework to make relevant knowledge available on demand to employees (Garone and Buck, 2000). The prevalence of the Internet has also propagated the explosive growth of web-based environments in enterprises. CORBA has distinct advantages in such an environment. It is easy to use -- design and code without explicit knowledge of the communication mechanism; it is object-oriented -- since distributed systems have a large number of states that need to be monitored, encapsulation and the use of exceptions can significantly increase the ease of managing the states; it optimizes communication traffic -- its objects are passed by reference rather than by value; it has a set of common services -- including object initiation and a naming service; and it is platform independent (Kashima, 1999). In addition, CORBA provides a mechanism for reuse. If problem-solving methods and knowledge bases are reusable components, then this should lead to a number of benefits for the developer. Since knowledge-base construction is expensive and time-consuming, the reuse of an existing knowledge base, even if it requires adaptation or augmentation, will lead to significant savings. A unified point of access that connects the intranet to the corporate business by integrating all of an enterprise's applications and back-end systems provide access to employees, which enhances productivity.

The integrated system uses the CORBA services to meet the above requirements with a central naming service as the discovery mechanism. The naming service allows clients to find objects within a distributed environment based on abstract/natural names.

Figure 2: Simplified View of the Integrated System

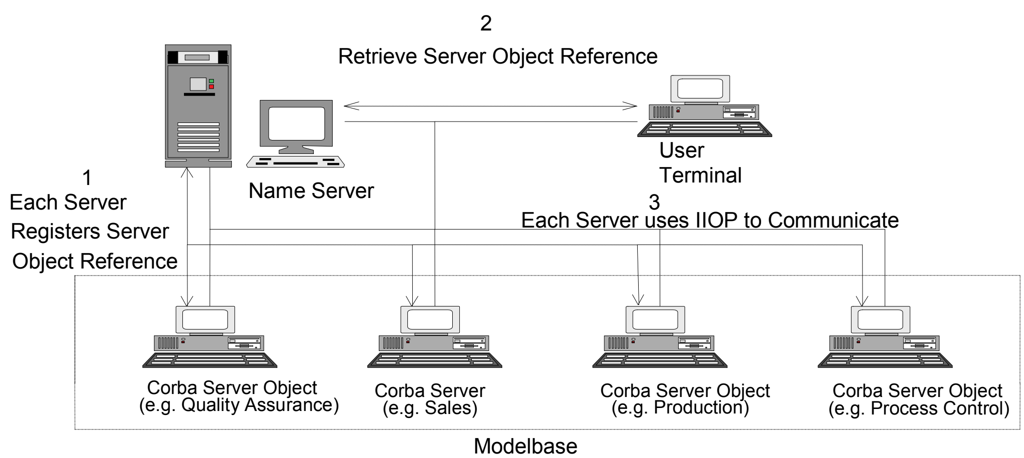


Figure 3 shows a server object registers with the naming service by invoking a bind to associate a logical name with an object reference. The naming server adds this object reference/name binding to its namespace database. A client application looks for a server object by invoking resolve to obtain an object reference with this name, and the client uses the object reference to call or invoke methods on the target object. Clients (user programs) use the Interoperable Object References (IOR) to communicate with server objects (component models and sub-programs).

The Model Base is a collection of different DSS components. A CORBA Interface Definition Language (IDL) defines the public interface between the application and each component model. This IDL is independent of the programming language in which it is implemented. It can also be used as a design tool for partitioning systems into components. The application communicates with the CORBA Naming Service to obtain the object reference for the component model and then binds to the object reference for the component model. The Internet Inter-Object Request Broker Protocol (IIOP) request to the component model contains the method call of the component model and the required parameters. Upon receipt of the parameters, the component model will process the request and return the results in the data type as defined in the IDL. The results are returned as an IIOP response to the application.

Ontology of the component models can be set up for the different component models of the organization-integrated system to inter-operate effectively. Ontology is a conceptualization of the different DSS components and their relationship. This ontology is implemented through the naming service. The role of the naming service is to provide an association between abstract names and CORBA objects. Structured tree type unique names must be assigned to various application components within the distributed environment. This assignment will also allow for flexibility in terms of navigation and configuration. The ontology and naming convention structure is illustrated in Figure 3.

Figure 3: A Component Model Aware Naming Context Graph (Ontology)

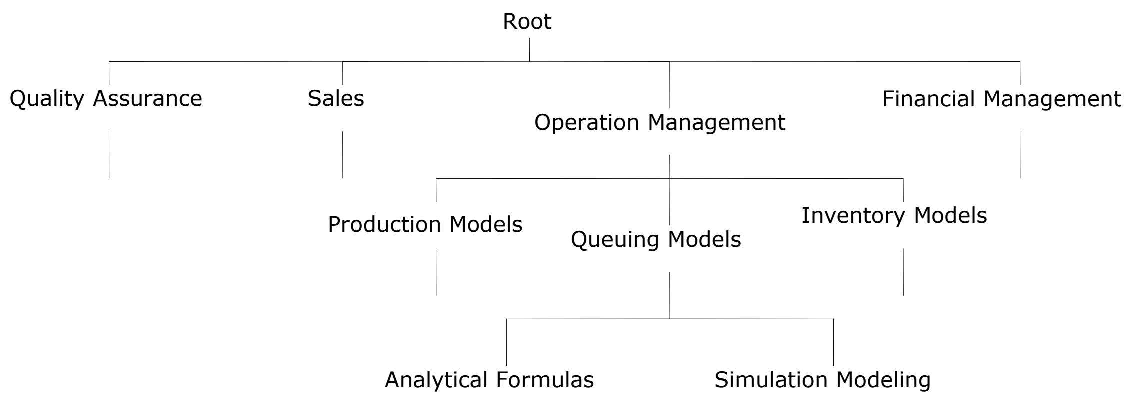


Figure 4 depicts a possible enterprise naming context graph. This graph allows for enterprise level services as well as subsystems and services that are specific to a particular department. This contextual graph will assist navigation to a CORBA object within a multi-departmental enterprise facility. Leaf nodes associate a name with a reference to a CORBA object.

This naming context graph includes a departmental context below the enterprise root context. This layering is one approach for handling multiple departments within the enterprise. When the entire component models within the enterprise is associated and named on this ontology (or naming context graph), any application object within the enterprise network can be invoked. This referencing leverages one of the advantages of CORBA -- location transparency.

CORBA provides the Naming Service as the discovery mechanism. Before an object or components can be accessed, a client must obtain a remote object reference. There are a number of standard specifications that have been developed in different contexts, which address the object discovery. The CORBA Naming Service provides a mapping between a name and an object reference. Storing such a mapping in the Naming Service is known as binding an object. Obtaining an object reference, which is bound to a name, is known as resolving the name. Names can be hierarchically structured by using contexts. Contexts are similar to directories in file systems and they can contain name bindings as well as sub-contexts. Object references are opaque data types, and their string form is a long sequence of numbers. When a service is restarted, its objects typically have new object references. However, in most cases clients want to use the service repeatedly without needing to be aware that the service has been restarted. The Naming Service solves these problems by providing an extra layer of abstraction for the identification of objects. It provides readable object identifiers for the human user. Users can assign names that look like structured file names, giving a persistent identification mechanism.

Objects can bind themselves under the same name regardless of their object reference. The typical use of the Naming Service involves object implementations binding to the Naming Service when they come into existence and unbinding before they terminate. Clients resolve names to objects, on which they subsequently invoke operations. Use of the naming service provides the following benefits: (a) it provides a central repository for object references accessible from applications, thus avoiding having to distribute them to clients via an alternative method, and (b) it allows clients to locate objects through standard names that are independent of the corresponding object references.

Thus, through the use of CORBA the distributed disparate "islands of automation" in an enterprise can be integrated (called IDSS) to leverage its corporate memory and expertise to increase the efficiency and effectiveness of business processes, resulting in better decision making, and in so doing maximize the enterprise's competitive advantage.

CONCLUSION

Therefore, CORBA can be used to implement the IDSS. A proof of concept experiment was conducted with data to validate the IDSS. The IDSS can improve decision-making outcomes relative to a "normal" decision support system for a semi-structured problem. The applied research carried out for this paper has contributed to the body of knowledge in enterprise systems integration, decision support systems and corporate memories. The contribution of this research is significant in a number of ways:

1. The utilization of information technology, decision support system and CORBA has significantly advanced the effectiveness of model management within decision support systems.
2. The proposed enterprise-wide, open, plug and play knowledge system architecture is a novel way to integrate problem-solving components.
3. The IDSS, through the integration of management science models with decision support concepts and delivering it through CORBA has shown that disparate knowledge and expertise can be delivered to the decision maker for the effective solution of decision problems.

There are bound to be some form of limitation in any research, as research cannot be conducted in an ideal environment. Thus, inherent in this research, there are some limitations: Even though the operating mechanism will be the same, only one application scenario was used in this evaluation, evaluating more scenarios will give a more "complete picture"; The scalability of the IDSS was not tested extensively; Evaluation of the application scenario was done under "laboratory" conditions instead of in a real-time enterprise; and Human subjects were not comprehensively used in the evaluation, however, students have tried the system and have provided some useful feedback.

All the limitations listed above can be overcome in future research. The recommended work for future research include the following: Cross-fertilizing the IDSS by testing more application scenarios to obtain results that are more complete; Porting the IDSS to the Internet and making it available to everyone. This will allow others to add more problem solving components to the system and (with more publicity) would also increase the number of users. These conditions will provide for scalability testing. This testing is expected to be more rigorous than that in an average real-time enterprise; Human subject testing can be done to determine the effects of the IDSS on the decision-making process and to determine if there is a co-relation to the decision-making outcomes. There are two types of users, the knowledgeable and the novice users. It is important to note that a knowledgeable user of a problem type can be a novice user of other problem types. A logging mechanism can be developed to capture users' keystrokes and mouse clicks but users' thought processes will still need to be captured through user feedback; Embedding intelligent agents in the IDSS user interface where the agent logs/watches and intervenes appropriately with questions and explanations (intelligent guidance); Embedding intelligent agents in the IDSS to personalize it so that it learns the users' preferences (intelligent assistant); Developing techniques or algorithms for the automatic division or decomposition of problems into smaller problems that can be easily solved and then integrate or re-assemble these solutions to solve the main problem by means of problem solving path construction and/or distributed problem solving; and Performing comparative studies of the performance of agent-based versus CORBA-based approaches for managing and utilizing corporate memories.

The scope of the work in the future research has been expanded significantly. This will allow for a more comprehensive study of corporate IDSS.

REFERENCES

Applegate, L.M., R.D. Austin and F.W. McFarlan (2005), *Corporate Information Strategy and Management: Text and Cases* (seventh ed). New York: Irwin/McGraw Hill.

Bhargava, H.K., Power, D.J., Sun, D. (2007) "Progress in Web-based decision support technologies," *Decision Support Systems*, Volume 43, Issue 4, Special Issue Clusters, August 2007, p. 1083-1095.

Bhargava, H. K. and Krishnan, R. (1993). "Computer-aided Model Construction," *Decision Support Systems*, 9(1), p. 91-111.

Bhargava, H. K., Krishnan, R. and Muller, R. (1997). "Decision Support on Demand: Emerging Electronic Markets for Decision Technologies," *Decision Support Systems*, Vol. 19, p. 193-214.

Bhargava, H. K., Krishnan, R. and Müller, R. (1997). "Electronic Commerce in Decision Technologies: A Business Cycle Analysis," *International Journal of Electronic Commerce*, 1(4). p. 109-127.

Bhargava, H.K. and Power, D.J. (2001). "Decision Support Systems and Web Technologies: A Status Report," *Proceedings of Americas Conference of the Association of Information Systems*, Boston, MA, August 2001.

Bhargava, H.K. and Norris, R. (1996). *D-Lib Magazine*, June 1996. http://www.dlib.org/dlib/june96/dnet/06norris.html

Bih, J. (2006). "Service Oriented Architecture (SOA): A New Paradigm to Implement Dynamic E-Business Solutions," *Ubiquity*, Volume 7, Issue 30, August 8-14, 2006.

Brose, G., Vogel, A. and Duddy, K. (2001) Java *Programming with CORBA*, Third Edition, Wiley, John & Sons, Incorp. 2001.

Burstein, Frada and Holsapple, Clyde W. (2008) "New Horizons at the Core of Information Systems: Decision Support System Advances," *Information Systems and E-Business Management*, Volume 6, Number 2, March 2008, p. 109-115.

Carlsson, C. and Walden, P. (2000). "Intelligent Support Systems - The Next Few DSS Steps," *Human Systems Management*, Vol. 19, 2000, p. 135-147.

Carlsson, C., Turban, E. (2002), "DSS: directions for the next decade," *Decision Support Systems*, Volume 33, Issue 2, June 2002, p. 105-110.

Chari, K. (2003). "Model composition in a distributed environment," *Decision Support Systems*, 35, p. 399–413.

Clarke, J., Stikeleather, J. and Fingar, P. (1996) *Distributed Object Computing For Business*, The Next Generation Computing Series, The Technical Resource Connection, Inc.

Cohen, M., Kelly, C.B. and Medaglia, A.L. (2001). "Decision Support with Web-Enabled Software," *Interfaces*, March-April, 2001.

Dabke, Padmanabh (1999) "Enterprise Integration Via CORBA-based Information Agents," *IEEE Internet Computing*, September-October 1999, pp. 2-10.

De Jong, I., et al. (2002). "Web Services/SOAP and CORBA," Compilation from comp.object.corba user group, April 27, 2002.

Eom, Sean B. (2002). "Extended Enterprise Decision Support Systems: A New Frontier in the Internet Age," *Proceedings of the IFIP TC8/WG8.3 Conference,* July 4–7, p. 726–739.

Fenn, J., et al. (2008). "Hype Cycle for Emerging Technologies, 2008," *Gartner Strategic Analysis Report*, 9 July 2008.

Firestone, J.M. (1997) "Distributed Knowledge Management Systems (DKMS): The Next Wave in DSS," *White paper No. 6*, Executive Information systems, Inc.

Firestone, J.M. (1999) "Enterprise Knowledge Management Modeling and Distributed Knowledge Management Systems," *White paper No. 12*, Executive Information systems, Inc.

Forgionne, G.A. (1991a). "Decision Technology Systems: A vehicle to consolidate decision-making support," *Information Processing and Management*, 27(6), p. 679-697.

Forgionne, G.A. (1991b). "Decision Technology Systems: A step towards complete decision support," *Information Systems Management*, Fall 1991, p. 34-43.

Gachet, Alexandre (2003) "A Jini-based software framework for developing distributed cooperative decision support systems," *Software, practice & experience*, Volume: 33, Issue 3.

Gachet, Alexandre (2002). "A New Vision for Distributed Decision Support Systems," *Proceedings of the IFIP TC8/WG8.3 Conference,* July 4–7, p. 341–352.

Garone, Steve and Buck, Neena. (2000) "Capturing, Reusing and Applying Knowledge for Competitive Advantage," *White paper*, International Data Corporation, Farmingham, MA 01701.

Gray, N.A.B. (2003) "Comparison of Web Services, JAVA-RMI, and CORBA Service Implementations," *Technical Report*, School of Information Technology and Computer Science, University of Wollongong, Australia.

Goul, Michael and Corral, Karen (2007) "Enterprise model management and next generation decision support", *Decision Support Systems*, Volume 43, Issue 3, Integrated Decision Support, April 2007, p. 915-932.

Holsapple, C.W. (2001) "Editorial. Knowledge management support of decision making," *Decision Support Systems*, 31 (2001), pp. 1-3.

Hummingbird (2001) "Enterprise Information Portals: Enabling Knowledge Management in Today's knowledge Economy," *White paper*, Hummingbird Ltd.

Kashima, Hiroshi. (1999) *An Approach for Constructing Web Enterprise Systems on Distributed Objects*. Technical report, IBM Japan Systems Engineering Co., Ltd.

Keen, P.G.W. (1987) "Decision Support Systems: the next decade," *Decision Support Systems*, 3(1987), pp. 253-265.

Keen, PGW and Morton, MSS. (1978) *Decision Support Systems: An Organizational Perspective*, Addison-Wesley, 1978.

Krishna, A. S. , Schmidt, D. C. , Klefstad, R. and Corsaro, A. (2003), "Real-time CORBA Middleware," *Middleware for Communications*, edited by Qusay Mahmoud, Wiley and Sons, New York.

Linden, Alexander and Fenn, Jackie (2003) "Understanding Gartner's Hype Cycles," *Gartner Strategic Analysis Report*, R-20-1971, 30 May 2003.

March, Salvatore T. and Hevner, Alan R.(2007) "Integrated decision support systems: A data warehousing perspective," *Decision Support Systems*, Volume 43, Issue 3, Integrated Decision Support, April 2007, p. 1031-1043.

McCoy, D.W.; Pezzini, M., Schulte, R.W., Thompson, J., Lheureux, B., Sinur, J., Kenney, F., and Natis, Y.V. (2003) "Hype Cycle for Application Integration and Platform Middleware," *Gartner Strategic Analysis Report*, R-20-0814, 30 May 2003.

OMG (2004) *CORBA 3.0 Specifications* (http://www.omg.org/technology/documents/corba_spec_catalog.htm)

Orfali, R. and Harkey, D. (1998). *Client/Server Programming with Java and CORBA,* Second Edition, Wiley, John & Sons, Incorp. 1998.

Porter, M. E. (1990) *The competitive advantage of nations*. New York: Free Press, 1990.

Powell, R. (2001). "DM Review: A 10-Year Journey," *DM Review*, February 2001.

Power, D. J. (1997). "What is a DSS?" *DSstar*, The On-Line Executive Journal for Data-Intensive Decision Support, October 21, 1997: Vol. 1, No. 3.

Power, D. J. (1998). "Finding DSS Resources on the World-Wide Web," *Journal of Decision Systems*, vol. 7, 1998, p. 7-20.

Power, D. J. (1998). "Web-based Decision Support Systems," *DSstar*, The On-Line Executive Journal for Data-Intensive Decision Support, August 18 and 25, 1998: Vol. 2, Nos. 33 and 34.

Power, D. J. (1999). "A Brief History of Decision Support Systems," http://dssresources.com/history/dsshistory.html, version 1.2, March 22, 1999.

Power, D. J. (2000a). "Web-Based and Model-Driven Decision Support Systems: Concepts and Issues," Model-Driven and Web-Based Decision Support Systems Mini Track, AMCIS 2000, *Proceedings of the Americas Conference on Information Systems*, Long Beach, California, August 10th - 13th, 2000.

Power, D. J. (2000b). "Supporting Decision-Makers: An Expanded Framework," http://dssresources.com/papers/supportingdm/sld001.htm, version 1.0, December 15, 2000.

Power, D. J. and Kaparthi, S. (1998). "The Changing Technological Context of Decision Support Systems," In Berkeley, D., G. Widmeyer, P. Brezillion & V. Rajkovic (Eds.) *Context-Sensitive Decision Support Systems*. London: Chapman and Hall, 1998.

Rabarijaona, Dieng, Corby, Ouaddari. (2000) "Building and searching a XML-based Corporate

Memory," *IEEE Intelligent Systems Special Issue Knowledge Management and Internet,* p. 56-64.

Schmidt, D.C., Vinoski, S. (2004), "The corba component model: Part 1, evolving towards component middleware," *C/C++ Users Journal*

Shim, J. P., Warkentin, M., Courtney, J.F., Power, D.J., Sharda, R., Carlsson, C. (2002). "Past, present, and future of decision support technology," *Decision Support Systems*, Volume 33, Issue 2, June 2002, p. 111-126.

Simon, H. (1977). *The New Science of Management Decision*. Englewood Cliffs, NJ, Prentice Hall.

Smirnov, A. (2006) "Integrated Technology for Distributed Operational Decision Support Systems," IEEE*/SMC International Conference on System of Systems Engineering.*

Steels, L. (1993) "Corporate Knowledge Management," In: Barthès, J.-P. (ed) *Proceedings of the first International Symposium on the Management of Industrial and Corporate Knowledge, ISMICK'93*, p. 9-30

Vinoski, S. (2003), "Integration with Web Services," *IEEE Internet Computing*, vol. 7, no. 6, pp. 75–77

Vinoski, S., and Lea, D. (2003). "Middleware for web services," IEEE *Internet Computing*, (January–February 2003).

Vinoski, S. (2002), "Web Services Interaction Models—Part 1: Current Practice," *IEEE Internet Computing,* vol. 6, no. 3, May/June 2002, pp. 89–91.

Vinoski, S. (1997) "CORBA: Integrating Diverse Applications Within Distributed Heterogeneous Environments," *IEEE Communications Magazine*, February 1997.

Voelter, M., Kircher, M. and Zdun, U. (2004) *Remoting Patterns: Foundations of Enterprise, Internet, and Realtime Distributed Object Middleware*, John Wiley & Sons, 2004.

Walther, E., Eriksson, H., and Musen, M.A. (1992) "Plug and Play: Construction of task-specific expert-system shells using sharable context ontologies," *Proceedings of the AAAI Workshop on Knowledge Representation Aspects of Knowledge Acquisition*, San Jose, CA, 191-198.

Whitehead, K. (2002), *Component Based Development: Principles and Planning for Business Systems*. Addison-Wesley

Zhang, Shifeng and Goddard, Steve (2007) "A software architecture and framework for Web-based distributed Decision Support Systems," *Decision Support Systems*, Volume 43, Issue 4, Special Issue Clusters, August 2007, Pages 1133-1150

Zahavi, R. (1999). *Enterprise Application Integration with CORBA Component and Web-Based Solutions,* John Wiley & Sons, Incorp. 1999.

BIOGRAPHY

Benjamin Khoo completed his Ph.D. (Information Systems) at the University of Maryland, Baltimore County. He is a member of two honor societies and was awarded the Phi Kappa Phi Dissertation Research Grant. He has published regularly in information systems journals. Prior to becoming an academician, he was a member of the Technical Staff (Software Engineer) of a large telecommunication Corporation.

Guisseppi Forgionne is a Professor of Information Systems, University of Maryland Baltimore County. He teaches Decision Support Systems and Decision Science courses, and had received a number of industrial grants for his research work. He has also supervised many doctoral dissertations and publishes often in the major information systems journals.

Peter Harris is a professional accountant and is an Associate Professor of Accounting at The New York Institute of Technology. He completed his MBA degree at Columbia University in New York City. He has also worked for Ernst and Young LLP and is a member of several professional organizations.