KEMI-TORNIO UNIVERSITY OF APPLIED SCIENCES

Enhancing Capabilities of Intelligent Decision Support Systems with Semantic Web Technologies

Viet T. Nguyen

Bachelor's Thesis of Degree Program in Business Information Technology

Bachelor of Business Administration

TORNIO 2012

ABSTRACT

Nguyen, Tuan Viet 2012. Enhancing Capabilities of Intelligent Decision Support Systems with Semantic Web Technologies. Bachelor's Thesis. Kemi-Tornio University of Applied Sciences. Department of Information Processing. Pages 68.

This research is focused on the area of Intelligent Decision Support Systems (IDSSs) and Semantic Web technologies to provide insight into their potential integration. This research aims to propose ways in which Semantic Web technologies can be integrated with Intelligent Decision Support components to enhance their effectiveness and efficiency.

Inspired by the intelligent aspects of the Semantic Web, this work was formed. IDSSs were chosen as the field to apply Semantic Web technologies. To accomplish the aim of this work, the backgrounds of the Semantic Web and IDSSs were studied. Afterwards, potential Semantic Web technologies that can be integrated into IDSSs environments are proposed. Finally, this work gives implications of this integration and suggests directions for future research.

On the basis of the research questions listed and the area of research, exploratory research methodology was chosen. Exploratory research is relevant for this research because it enabled studies into the Semantic Web and IDSSs technologies to be conducted. Subsequently, assumptions and theories to their integration could be given. This research explores and provides suggestions for the integration, not concrete groundings for future applications. This work makes extensive use of secondary sources such as scientific journals and books from established authors.

The outputs of this work are ideas in which Semantic Web technologies can aid IDSSs, as well as a conceptual model of an IDSS environment incorporating SW technologies.

Keywords: decision support systems, intelligent DSS, semantic web, knowledge-based, semantic web technologies, web services.

CONTENT

ABSTRACT

FIGURES

FIGURES

1 INTRODUCTION

Decision making is part of our everyday lives and it affects the operations of major fields such as business, healthcare, military, and government. In these fields particularly, the effectiveness of the decisions made is vital as people's lives and their properties are at risk if wrong decisions are made. Nevertheless, not all of the decisions faced in these fields are a matter of life and death, and decision making in most cases involves planning ahead and ruling out alternatives based on given information.

Decision making can be defined as "a process of choosing among two or more alternative courses of action for the purpose of attaining one or more goals" (Turban & Sharda & Delen 2011, 41). Information technology (IT) has penetrated many aspects of human activity today, and in order to improve on these activities, strengthening their integration with IT should be a primary focus. Decision making is one of those activities affected by IT, and Computer–aided decision making was proposed in the 1960s by researchers through studies and experiments (Power 2007). It was in the early 1970s, however, that Scott-Morton gave specific meaning to decision support systems (DSSs) in which he defines them as "interactive computer-based systems, which help decision makers utilize *data* and *models* to solve unstructured problems" (Gory & Scott-Morton 1971 as cited by Turban et al. 2011, 16 original emphases). The purpose of DSSs is to help decision makers throughout the whole decision-making process (Turban et al. 2011, 75). Since the HTML 2.0 specification was introduced around 1995, Web-based DSSs started to evolve (Power 2007). Web-based DSS was since acknowledged by a number of professionals and academics as a serious platform for implementing all types of DSSs (Berners-Lee 1996 as cited by Power 2007). Internet technologies are evolving and DSSs should be perfected to meet the standards of future Internet technologies.

As the Internet or Web grows, the amount of information that is stored in it increases in size. Search engines today serve the purpose of finding all the information that is available on the Web. However, the search engines only tend to find the metadata, which is data about data, that are defined within the web pages and resources of the Web. Current computer systems that run these search engines do not understand the meaning of neither the metadata nor the web pages and resources that the metadata defines. In dealing with this problem, Berners-Lee et al. (2001 as cited by Lu & Roberts & Lang & Stirling & Madelin 2006, 323) developed the idea of the Semantic Web to aid with Web semantic interoperability. Semantic Web in its envisioned form provides the ability to tag all content on the Web, describe the contents' meaning and give them semantic meaning (Bose & Sugumaran, 222). The term "semantic" deals with the meaning of words; Semantic Web has technologies such as Ontology, Resource Definition Framework (RDF) and RDF schema (RDFS) in an attempt to describe the semantic meaning of Web content (Bose & Sugumaran 2007, 224).

As the Web is a predominant deployment platform of DSSs and Semantic Web technologies are increasingly being developed to become the next Web standard, studies to explore the possible use of Semantic Web technologies within DSS environments are relevant and innovative. Previous studies on the integration between DSSs and the Semantic Web included such works as proposing a model to enhance web-based DSS environments through Web services by means of Semantic Web technologies (Bose & Sugumaran 2007) and undertaking exploratory research with the development of ontologies to be used by IDSSs for UK railway systems (Lu et al. 2006). Bose and Sugumaran's (2007, 227-231) work, whose aims are closely related to the objective of my research, focuses on using Semantic Web technologies such as: (1) ontologies to provide context for data and support discovery of Web services, modeling, assembly, mediation, and semantic interoperability, (2) ontology mapping and matching to interoperate distributed sources in the Semantic Web, (3) Semantic Knowledge Management to manage semantic knowledge acquisition, (4) semantic user interface aided by knowledge agents and ontology-based information, and (5) Semantic Web services that extend typical Web services with meaningful representation of information of Web services.

1.1 Motivation

Further study into the integration of Semantic Web technologies into IDSS environment, especially at the level of direct integration into IDSSs sub-systems, is needed to gain a more comprehensive understanding of how to enhance intelligence in IDSS. Previous research on this topic, such as that of Bose and Sugumaran's (2007, 15-16) work, prompted the need to expand to other aspects of Semantic Web technologies in order for the integration to work. These aspects include acquiring and managing knowledge, developing ontologies, mapping and matching of ontologies, and adding semantics to DSSs' functionalities. Delen and Pratt's (2006, 334) research on an industrial IDSS commended future researchers to build on their existing framework to support for a wider range of application domains. The conceptual model proposed in this research is based on Delen & Pratt's framework, as well as based on a few other notable frameworks that are described in Chapter 6. With the assumption that the Semantic Web is in full deployment, this research will attempt to explain how Semantic Web services and tools can be beneficial to IDSSs environments.

The software field is exploring the use of Semantic Web technologies in IDSSs because they provide an attractive, platform-neutral, application-neutral, Web environment that operates on top of the existing Web without having to modify it (Bose & Sugumaran 2007, 211). Semantic Web is currently in a developmental stage and has only seen few adopters of its technology, thus there is a lot of room for improvement (Herman 2009, 2). Also, IDSSs are increasingly being developed for the Web (Bose & Sugumaran, 223; Lu et al. 2006, 322) and Semantic Web is expected to integrate with the existing Web (Herman 2009, 3). Therefore, it is likely that in the future IDSSs would conform and evolve to meet the standards set by Semantic Web. It is therefore relevant that research in an attempt to combine Semantic Web and IDSSs should be performed. A decision support system combining the power of Semantic Web ontology matching and mapping, search engine, Web Service registries and Web mining techniques would see a notable increase in performance and reliability in its decision making.

The need for integrating Semantic Web technologies into IDSSs can also be emphasized through supply and demand. For example, a decision maker may need to access resources that are located in various parts of an organization which are separated geographically and located throughout vast amounts of databases. These resources will be hard to find without meaningful metadata that define them. Therefore, IDSSs demand services and tools which can facilitate the data mining process as well as efficiently and effectively organize the data themself. This is where Semantic Web technologies come in. Semantic Web provides means to integrate heterogeneous systems across organizations or organizational units in a meaningful way using such technologies as ontology, knowledge representation and intelligent agents (Bose & Sugumaran 2007, 227).

1.2 Objectives

With the background of this research briefly discussed and the need to carry out further research motivated, the objectives of this research are to study the Semantic Web technologies that can be integrated into IDSSs. Upon the findings of several researches, this thesis forms a conceptual model of an integrated IDSS environment within the Semantic Web. The objectives of this research stemmed from the fact that as Internet technologies rapidly develop DSSs are increasingly used in many organizations (Shim et al. 2002 as cited by Lu et al. 2006, 322). Also, since the volume of data and human-centered information available to decision-makers increases rapidly, the need to represent this information in software-process able formats becomes more apparent (Bose & Sugumaran 2007, 223). Therefore, to achieve these objectives, this work will suggest ways in which IDSSs can harness the Semantic Web technologies by first studying their internal workings and then draw out possible solutions for their integration.

1.3 Structure of the thesis

This chapter has presented a brief introduction to the field, topic, objectives and motivation of the research. The rest of the thesis is organized as follows. The research questions and research methodology are discussed in Chapter 2. An introduction to the two disciplines, Semantic Web and IDSSs, is given in Chapters 3 and 4. Chapter 5 analyses the Web Services technology to give suggestions on how it may be combined with the Semantic Web to create future IDSSs. Chapter 6 proposes ways to integrate Semantic Web technologies into an IDSS environment with a conceptual model being presented as the primary output of this research. Finally, Chapter 7 presents the conclusions, discusses the results of this work, and suggests directions for further research.

2 RESEARCH OBJECTIVES, QUESTIONS AND METHODOLOGY

The topic of this research is to focus on the concepts and technologies of IDSSs and the Semantic Web. The main objectives of this work are to study these technologies in details, and to propose and describe an integration of the Semantic Web technologies and IDSS environments. To achieve these objectives, this thesis gives a brief historic overview of DSSs and Semantic Web, noting in particular their theoretical development and applications. Secondly, this thesis describes the core technologies that make up IDSSs and the Semantic Web as well as an outline of their collaborative functionalities. Furthermore, a few noteworthy examples are given to illustrate the concepts of IDSSs and the Semantic Web. Based on these steps, this thesis draws connections and a possible integration between IDSSs and the Semantic Web. The main output of this research is the proposition for further development and mutual integration of the two technologies.

This chapter defines the research questions of this study and the research methodology.

2.1 Research questions

Three main research questions are defined to achieve the objectives of this thesis and the steps necessary to answer them are detailed below.

1. What are the concepts behind Semantic Web and IDSSs? What underlying technologies are they comprised of?

   To understand the potential ways Semantic Web and IDSSs can be integrated, the fundamental concepts of these two disciplines must be understood. First, a brief history of these technologies is provided and examples are given to illustrate their applications and development. Following the historical overview, definitions of the technologies of Semantic Web and IDSSs are given as well as explanations of any relationships these definitions may have with one another. Lastly, the most important underlying technologies governing IDSSs and Semantic Web are studied in details.

2. How can the technologies of Semantic Web and IDSSs be integrated?

   Web-based DSSs have been developed since HTML 2.0, further extending the capabilities and deployment of computerized decision making systems (Power 2009). This research will show how IDSSs on the web can be enhanced by Semantic Web Services and applications and benefits from the potential data that can be retrieved from the Internet. Thus, this research will take a step forward into a possible future and assume that the Semantic Web is already deployed and that existing IDSSs environments are trying to harness this technology. This study will propose a way how the Semantic Web technologies can be incorporated into IDSSs environments and in what level. An analysis of the advantages and disadvantages of this incorporation will be made.

3. How can Semantic Web applications and technologies be developed to enhance IDSSs environment?

   Several methods in the integration of Semantic Web technologies and DSSs are proposed in this research: (1) making Semantic Web Services and applications easily available for IDSSs, (2) extending IDSSs' knowledge base and ontological resources, (3) using Semantic Web tools to support development of IDSSs technologies, and (4) exploring possible DSSs technology involving mobile technology. Furthermore, future research directions are given to improve and analyze the collaboration of the Semantic Web and IDSSs technologies.

2.2 Research methodology

This research involved mostly exploratory research based on the literature. The topic of this research is relevant to exploratory research due to its novelty. Intelligent Decision

Support Systems and the Semantic Web by themselves are not new concepts, but their combination requires further research and development (Bose & Sugumaran 2007, 235). The topic of this study therefore involved mainly theoretical thinking and reasoning. However, practical adaptations of the outputs of this research are possible. Literature review was the main method in this research as most practical developments involving IDSSs and Semantic Web are either closed systems or available only through user interfaces.

According to McDaniel Jr. and Gates (2010, 43)**,** the purpose of an exploratory research is "to obtain greater understanding of a concept or to help crystallize the definition of a problem. It is also used to identify important variables to be studied. Exploratory research is preliminary research, not the definitive research used to determine a course of action."

Therefore, exploratory research is relevant because this work was able to explore technologies of the disciplines IDSSs and Semantic Web and give suggestions for their integration. Discussions with the thesis supervisor also accompanied this exploratory research work. However, due to the nature of the topic of this research, literature review was the most suitable technique for this exploratory research. Secondary sources including books, reports and articles were used. Some tertiary resources such as survey reports and bibliographies of previous research were also used. The overall parameter for finding and using literature sources is that they should be relevant, current and published by reputable authors and publishers. These sources are to pertain to disciplines of DSSs and the Semantic Web technologies. Most of the sources used for conducting this research were mainly searched from the Internet due to the lack of relevant printed material in the library of Kemi-Tornio University of Applied Sciences (KTUAS). These Internet sources include all external electronic libraries that KTUAS has a subscription for.

Exploratory research provided an insight into current and emerging IDSSs and Semantic Web technologies and at the same time uncovered possible integration of the two disciplines.

3 DECISION SUPPORT SYSTEMS

In this chapter, a detailed analysis on how DSSs are composed and how they function is given. This chapter goes through the history and the usage of DSSs, explains the taxonomy of DSSs, defines what IDSSs are and how these are built and, finally, lists applications of DSSs.

3.1 DSSs to support decision makers

DSSs are a broad category of information systems for informing and supporting decision makers. DSSs are intended to improve and speed-up the processes by which people make and communicate decisions. (Power 2000, 6.) The history of the development of DSSs began more than 40 years ago. This history is not linear but rather based on multiple simultaneous research work. Initial development of a computer based decision system started during the mid-1960s involving experimental studies and field research**.** During the past decades, actual practical development in management decision systems helped business managers in their decision making. (Power 2007.) In the 1980s, the theory development of DSSs started to form, and researchers regarded DSSs to be a new class of information systems (Power 2009, 9). By the mid-1990s, the modern era of DSSs began with the Web being the main development environment, enabling the creation of revolutionary user interfaces (Power 2009, 12).

Since the emergence of the HTML 2.0 specification, DSSs technologies reached a turning point, and from here the way DSSs are designed and built, and their functions have been significantly changed (Power 2007; Turban et al. 2011). Within the Web environment, DSSs are given access to the huge collection of databases and models that the Web offers. Large database vendors such as IBM and Microsoft provide Web interfaces for accessing their databases, and even legacy databases are available through the Web. The Web is also used as a medium for communication and collaboration between distant group decision makers. A large collection of DSSs software is also available for download or purchase from Application Service Providers and various other online software vendors. Moreover, software development tools such as Java, PHP and the .NET framework enables access to the Web for DSSs developers, thus providing means to interface with databases and models. (Turban et al. 2011, 108.)

DSSs are built to support decision makers in all phases of decision making. To understand the extent of this support, knowledge of the decision making process must be attained before DSSs developers are to make fully functional DSSs. The first phase of the decision making process is the intelligence phase, and this is where the decision maker examines reality, identifies and defines the problem, and establishes the concept of *problem ownership*. The second phase, the design phase, involves the collection of data and the formation of models which simplifies the problem. From these models, decision makers are presented with alternative courses of action that result in alternative solutions to the problem. The choice phase includes the selection of one of those alternative solutions that was derived from the models. A test is conducted to verify the solution before implementing it to see if the initial problem is solved. Successful implementation results in the problem being solved, while failure indicates the opposite and the decision making process is iterated. (Turban et al. 2011, 43-46, 75.)

3.2 Taxonomy of DSSs

There have been two prominent kinds of taxonomies that attempt to categorize and further define the wide variety of roles DSSs could play. One was that of Steven Alter in which he defines seven types: file drawer systems, data analysis systems, analysis information systems, accounting and financial models, representation models, optimization models and suggestion models (Alter 1990 as cited by Power 2000, 6). The first three types can be broadened to be termed data-oriented or data-driven, the second three types to be termed model-oriented or model-driven and the last type can also be termed intelligent or knowledge-driven (or knowledge-based) DSS (KD-DSSs) (Power 2000, 7). These three broadened types, in addition to document-driven and communications-driven DSSs incorporate the second type of taxonomy proposed by Power. This taxonomy served as a basis for conducting this research.

Power (2002 as cited by Turban et al. 2011, 79) categorized DSSs into five main areas which were later adopted by the AIS SIGDSS. He categorizes DSSs into 5 types as follows: communications-driven, data-driven, document-driven, knowledge-driven and model-driven decision support systems (Power 2000, 17). There is also a sixth type of category, called the compound DSSs, which is an incorporation of two or more categories of DSS.

Communications-driven and group DSSs make it possible for multiple decision makers to collaborate and make decisions. These decision makers generally are geographically dispersed, which is one of the needs for this type of DSSs to exist. This category of DSSs typically uses communication standards and technologies to enable group members to reach and collaborate with each other. Almost all DSSs that involve group work fall into this category. Google Apps, a cloud computing service, can be seen as a communications-driven and group DSS because it involves offering Web services to business in order to save time and money to group-related work. More generally, communications-driven and group DSSs support meetings, design collaborations, and even supply chain management (Turban et al. 2011, 80). Thus, Communications-driven and group DSSs enable decision makers of various expertise and skill level to work on a problem area and draw out careful and comprehensive consensus to the decision making process.

Data-driven DSSs are DSSs that retrieve and manipulate data to present the manipulated information to decision makers. This type of DSSs relies on databases in their structure. Early data-driven DSSs use the relational database configuration. The information handled by relational databases tends to be voluminous, descriptive, and rigidly structured. A database-oriented DSS features strong report generation and query capabilities. Because of this, most tools that feature this application are marked under Business Intelligence (BI) umbrella. (Turban et al. 2011, 80.)

Document-driven DSSs rely on documents and texts which are formatted textually, verbally or visually. Most knowledge management systems (KMSs) fall into these categories. These DSSs essentially include all DSSs that are text-based. (Turban et al. 2011, 80.)

Knowledge-driven DSSs use Artificial Intelligence (AI) techniques and apply knowledge technologies to help with decision making. The terms knowledge-driven DSSs and IDSSs are used interchangeably (Power 2000a, 7; Michalewicz et al. 2005, 44; Turban et al. 2011, 80). IDSSs incorporate such AI techniques as Artificial Neural Networks (ANNs), Expert Systems (ESs), Intelligent Agents (IAs) and various other numerical techniques. These techniques are explained in more detail in section 3.4. More information on Knowledge-driven DSSs or IDSSs is given in section 3.3.

Model-driven DSSs focuses on using optimization or simulation models to optimize on one more objectives. These DSSs emphasize activities in model formulation, model maintenance, model management in distributed computing environments, and what-if analyses. An example of a model-driven DSSs is Microsoft Excel, which includes dozens of statistical packages, a linear programming package, and many financial and management models. (Turban et al. 2011, 80.)

Compound DSSs are DSSs that include two or more of the prominent categories above. Examples include data-driven DSSs which can effectively utilize optimization models. Also, documents may serve useful in understanding how to interpret results of the outputted analyses from data-driven DSSs. (Turban et al. 2011, 81.)

3.3 Intelligent Decision Support Systems

IDSSs are a sub-group of DSSs and they are characterized by having AI to automate the decision making process of the decision making systems. In the 1990s, DSSs were further enhanced with components from artificial intelligence and statistics. This type of DSSs have names which had been used interchangeably, including management information systems, intelligent information systems, expert systems, management-support systems, and knowledge-based systems. (Michalewicz et al. 2005, 44.) IDSSs can also be viewed as knowledge-driven DSSs (Foster et al. 2005, 4). In view of a knowledge-driven DSS, an IDSS can be defined as "person-computer systems with specialized problem-solving expertise. The "expertise" consists of knowledge about a particular domain, understanding of problems within that domain and "skill" at solving some of these problems." (Power 2000, 11.) Thus, apart from using data and analytical models, IDSSs also use qualitative and judgmental knowledge that are derived from human experts. The skills required to solve problems may lie in the use of IAs, Intelligent Decision Support tools or Data Mining tools (Power 2000, 11; Wang 1997, 326). To answer the question of "what makes an IDSS intelligent?" a comprehensive analysis on knowledge-based or IDSSs, including the tools and technologies to create them will be explained.

IDSSs defer from a traditional DSSs in a number of ways. First, IDSSs allows support for a wider range of decisions and even those with uncertainty. Enhancing from traditional DSSs, IDSSs not only gives recommendations but also grades the level of confidence upon the recommendations. IDSSs can also perform in domains requiring expertise and give an impact analysis of any decisions made. (Turban & Aaronson 2001; Sauter 1997 as cited by Foster et al 2005, 4.) Further, IDSSs are advantageous over traditional DSSs in a number of ways. These advantages include: improved timeliness, consistency in decision, explanations and justification for recommended actions, management of uncertainty, and the formalization of organizational knowledge. (Marakas 1997 as cited by Foster et al 2005, 4.)

The fundamental principle behind IDSSs is that information collected through the system is intelligently analyzed and after which the system outputs intelligent decisions for IDSSs users. It is through the intelligent analysis of information that makes IDSSs intelligent. (Ryabov 2011.) Through discussion with Dr. Ryabov (2011), further analysis was made at how IDSSs are built. Figure 1 shows how an IDSS helps with the decision making process and depicts the core IDSSs components, including an interface for placing requirements of the decision, a knowledge system in the form of databases that are stored on the Internet and also within an expert repository, and a problem-processing system that intelligently analyses the information gathered from the knowledge system.

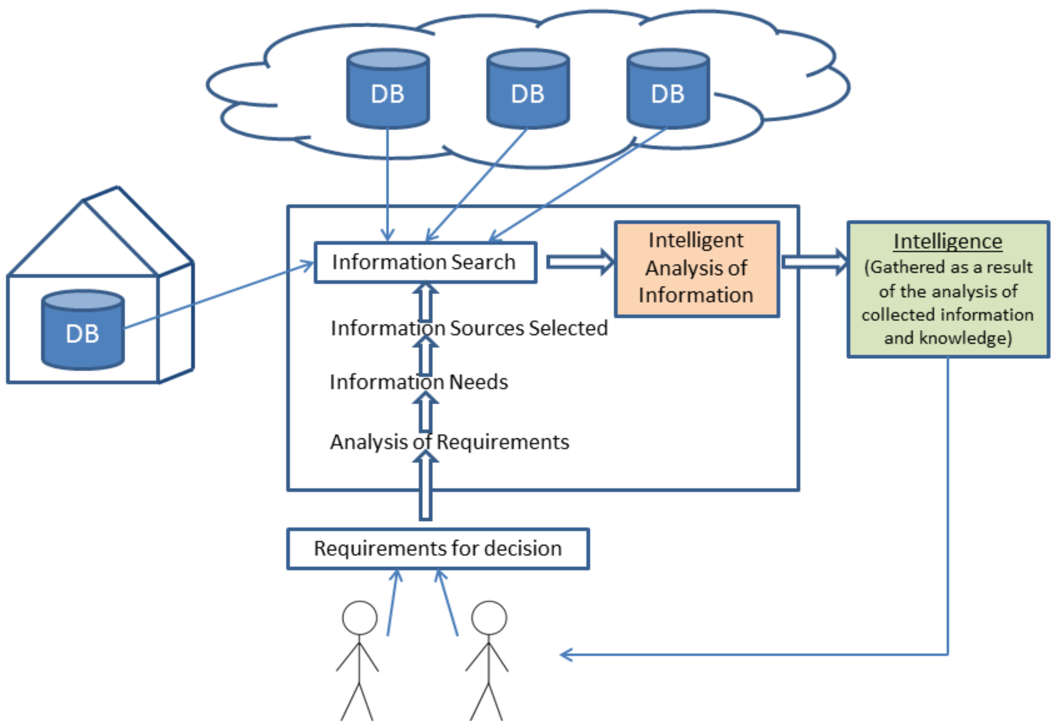

Figure 1. An Intelligent Decision Support System (Ryabov 2011)

In Figure 1, the "core" system of the IDSS includes the means for analyzing, processing and searching the information that is required to support a decision. Firstly, decision makers specify their decision requirements through the interface. The system will analyze these requirements, identify information to be gathered and select relevant information sources. After all the necessary information is gathered, the system will analyze the information using AI techniques, such as IAs or machine learning. The resulting analysis of the information will then form intelligence which is delivered back to the users. (Ryabov 2011.)

3.4 Core components of IDSS

Ryabov's structuring of IDSSs in the section above gave an overall idea of the decision process that happens within IDSSs. However, looking at IDSSs in terms of what they are composed of, Turban et al. (2011, 85-87) divided DSSs into 4 main types of components; a user interface subsystem, a model management subsystem, a data management subsystem, and a knowledge-based management subsystem. DSSs with the fourth component, the knowledge-based subsystems, have intelligence support to the other components, and as defined in the previous section such DSSs are called IDSSs. The four main components relationship with each other and with the decision maker can be seen in Figure 2.

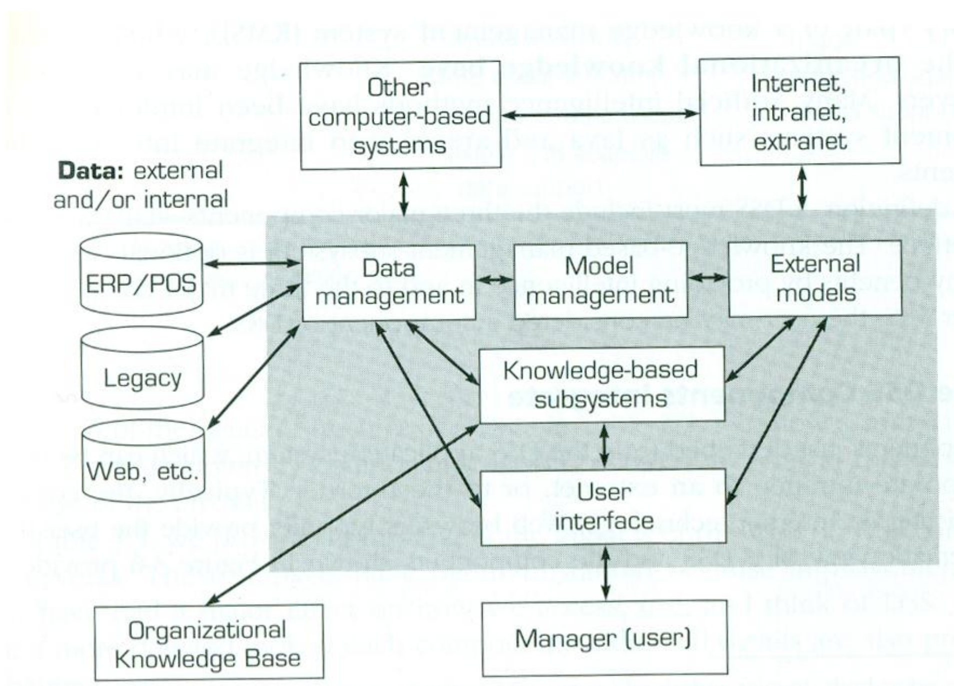

Figure 2. Schematic View of DSSs (Turban et al. 2011, 85)

In Figure 2, the Manager interacts only with the User Interface, which handles input/output functions from the other three components. The grey box in Figure 2 represents the whole system, while components outside of the box are external sources from which the DSSs acquire information. The components connect with each other via Internet technology. The

User Interface provides consistent Graphical User Interface (GUI) for the decision maker to interact with and take control of the DSSs. The user is also considered as a part of the system as research has shown that the effectiveness of DSSs is derived from the intensive interaction between the user and the computer. The data management subsystem includes database management system software to handle the databases containing relevant data for the DSSs. The databases are usually web servers, and can be internal or external to the DSSs. The model base subsystem has similar software called model based management system that includes financial, statistical, or other quantitative models and even modeling languages to provide the DSSs with analytical capabilities and suitable software management. The models run on application servers and can be either locally stored or retrieved from internet sources. The last component is the knowledge-based subsystem, which is unique to IDSSs, can provide intelligence support to the other components or act as an individual component. It can be connected with a local knowledge repository or with external Web servers. It can also be comprised of artificial intelligent methods and tools developed in Web development systems such as Java to assist the other components of the DSSs. (Turban et al. 2011, 86.)

3.5 Tools and technologies to support IDSSs

There are various AI techniques that can be applied in IDSSs. Crunk and North (2007, 64) assert that the incorporation of AI techniques are to be the natural development of DSSs in the coming future. These DSSs are characterized by robustness, ease of control, simplicity and completeness of relevant detail. These systems will be able to derive new knowledge based on existing knowledge, such as analyzing market trends and predict reliably product demographics. The best system will fit Little's and Alter's model in that it will be reusable across multiple organizations and companies (Power 2003 as cited by Crunk & North 2007, 64). This reusability is done through software or hardware framework conforming to Alter's model, which will then enable potential users of the system to take advantage of basic, generalized models that is common to a range of scenarios (Crunk & North 2007, 64).

As discussed in the previous sub section, IDSSs can support decisions even when the information given is under uncertainty. There are a number of AI techniques that can be used in order to handle such information. Before listing the different AI techniques that make an IDSS intelligent, the concepts as well as terms that revolve the term *uncertainty* must be understood in order to process information under uncertain circumstances.

Uncertainty is a term that can both be used as a generic term for imperfection in data, and as a term for a particular form of imperfect knowledge, that is, whether or not a statement is true. However, the term *imperfection* will be used henceforth in place of *uncertainty* to convey the generic meaning of imperfect information. (Parsons 1996; as cited by Ryabov 2002, 17.) Imperfection is distinguished by 5 types: uncertainty, imprecision, incompleteness, inconsistency and ignorance. In this definition, uncertainty is a situation when there is not enough certain information about the state of the world. (Parsons 1996; as cited by Ryabov 2002, 20.) These types of imperfection can stem from different sources, and within these sources particular applications are also determining factors (Ryabov 2002, 20). For example, different kinds of imperfection can result from unreliable sources such as faulty sensors and input errors, or from the inappropriate choice of representation (Motro 1993; as cited by Ryabov 2002, 21). Also, if data is recorded statistically, it is inherently uncertain. Measurement instruments might also introduce some imprecision in recorded data (Ryabov 2002, 21). Moreover, some information can be intentionally made uncertain for the reasons of security (Kwan et al. 1993 as cited by Ryabov 2002, 21).

There are two general classes of techniques for handling imperfection: numerical and symbolic (Ryabov 2002, 22). As the focus of this research is on the architecture of IDSSs and not on AI tools themselves, this report will only give a brief outline of the numerical technique, namely the probability theory and the possibility theory.

The probability theory is the oldest numerical technique for handling imperfection. In a concise form, “a probability measure is an estimate of the degree to which an uncertain event is likely to occur”. The probability theory has seen several extensions within artificial intelligence literature (Ryabov 2002, 23-24). In computing, the probability theory has been implemented into what is known as a probabilistic network, or also known as a Bayesian network or a causal network (Heckerman & Wellman 1995; Pearl 1987; Pearl 1988 as cited by Ryabov 2002). These networks are intended to represent and describe more efficiently probabilistic information using conditional probabilities together with further structural information (Ryabov 2002, 24).

The possibility theory is another numerical technique for handling imperfect data which emerged from the notion of fuzzy sets (Zadeh 1965 as cited by Ryabov 2002, 25) and was first introduced by Zadeh (1978 as cited by Ryabov 2002, 25). A fuzzy set “is a set whose membership is not absolute, but a matter of degree, for example the set of young people”. According to Parsons (1996 as cited by Ryabov 2002, 26), “there is a heuristic connection between possibility and probability”. Also, to elaborate to the statement by Parsons, Ryabov (2002, 26) reasons that “since if something is impossible, it is likely to be improbable, on the other hand, a high degree of possibility does not imply a high degree of probability, nor does a low degree of probability reflect a low degree of possibility”. Also, unlike the probability theory or the Dempster-Shafer theory - another numerical technique to handling imperfection - the possibility theory does not deal with propositions that are definitely true or false (Ryabov 2002, 26).

Apart from the numerical techniques explained above, there are several other techniques to help with decision support under imperfect conditions. Fuzzy logic is a technique that attempts to imitate the complex reasoning that is fundamental in the human ability to make rational decisions in an environment of imperfection (Crunk & North 2007, 64). Fuzzy Logic has only recently been implemented into. An example of fuzzy logic being applied in DSSs is AMOS, a probability-driven, customer-oriented DSSs for target marketing of solo mailings. (MaLec 2002 as cited by Crunk & North 2007, 64.) AMOS enables the user to perform market research on how customer behavior impacts new products (AMOS 2007 as cited by Crunk & North 2007, 64).

Another AI technique that helps DSSs in imperfect environments is ANNs. According to MaLec (2002 as cited by Crunk & North 2007, 64) ANNs are “are distributed information-processing systems that are important in modeling fuzzy and uncertain phenomena and in forecasting non-linear systems". MaLec (2002 as cited by Crunk & North 2007, 64) explained that ANNs can be helped to form models which could help analyze market segments, learn and store knowledge and capture markets with such tools as the Neural Network Model for Predicting Market Responses, and analyzing the market and learn from past experiences with Neural Network Modeling using neural networks combined with the PIMS (Profit Impact of Market Strategy) database.

ESs as mentioned earlier in the text are one of the main components of a KD-DSS (Klein & Methlie 1995 as cited by De Koch 2003, 7). ES are one of a variety of applications of Intelligent Systems. Intelligent Systems itself describe the applications of a much broader field, Artificial Intelligence. ESs provide the knowledge and reasoning for KD-DSS by having (1) a system which can simulate reasoning and (2) a system that can explain its reasoning and conclusions. (Turban et al. 2001 as cited by De Koch 2003, 7.) The role of ES can also be emphasized when Power (2000, 11) describes KD-DSSs as person-computer systems with specialized problem-solving expertise. ES is therefore ideal in assisting a decision-maker where expertise is required (Turban 1995 as cited by De Koch 2003, 7).

IAs have also been mentioned as the power that fuels IDSSs by such researchers as Foster et al. (2005) and Wang (1997). IAs are software agents that denote intelligent behaviors through a combination of software agents and intelligent systems (Wang 1997, 325). Agents, in a more general sense, can be described as "anything that can autonomously interact with its environment and an intelligent agent perceives its environment and makes informed decisions based on its perceptions and acts accordingly" (Kazma 1998; Vidal & Durfee 2003 as cited by Foster 2005, 2). An agent comprises of the following traits:

a. Autonomy: Agents operate without the direct intervention of humans.

b. Co-operativity: Agents co-operate with other agents towards the achievement of certain objectives.

c. Reactivity: Agents do not simply act in response to their environment, they are able to exhibit goal-directed behavior by taking the initiative.

d. Agents are able to travel through computer networks. An agent on one computer may create another agent on another computer for execution. Agents may also transport from computer to computer during execution and may carry accumulated knowledge and data with them. (Wooldridge & Jennings 1995 as cited by Wang 1997, 325.)

From what can be seen in the characteristics of an agent and in particular IAs, IDSSs can potentially reduce human-involvement and decision-making time if the system was managed through networks of agents. Figure 3 illustrates a multi-agent based DSSs.

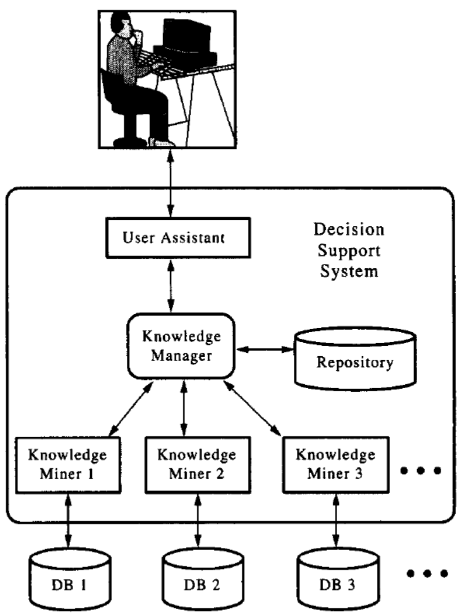

Figure 3. A multi-agent based DSS (Wang 1997, 326)

In Figure 3, there are three types of agents, or decision support agents: *Knowledge miner* discovers hidden data relations in information sources, *user assistants* act as the intelligent interface between the decision maker and the IA-assisted DSSs and a *knowledge manager* with repository support that provides system co-ordination and facilitates knowledge communication. (Wang 1997, 326-327.)

3.6 Applications of Decision Support Systems

This section deals with current applications of DSSs and possible applications in the future. This section details some common applications of DSSs, and is categorized into fields in which DSSs are applied.

3.6.1 Business applications

In marketing, ESs have been designed to store specific market knowledge from experts and to make that knowledge available for problem solving. These systems support marketing decision making. Applications of ESs for marketing decisions include STRATEX and COMSTRAT. (MaLec 2002 as cited by Crunk & North 2007.) In education, ES have been used to aid various educational disciplines such as computer animation, computer science, engineering, languages and business study. Most of these ESs have Intelligent Tutoring Systems (ITSs) which have techniques such as active hypertext and hypermedia. (Kiong & Rahman & Zaiyadi & Aziz 2005.)

ITSs embedded in ESs were used to teach engineering students to gain deep understanding of the fundamentals to be able to follow the more advanced topics in the engineering field. The ITS used a fuzzy rule based decision making system that would guide the ITS's behavior. For each student, this ES will draw the information regarding student's performance against the membership function for each topic, difficulty and importance level. (Kiong et al. 2005.)

IAs have played a significant role in online shopping with the comparison-shopping agent. The comparison-shopping agents "are web-based intelligence information systems that can collect product and service information- especially price-related information – from multiple online vendors, aggregate them, and then process them into value-added information for online shoppers to assist their decision making (Wan 2006, 147). The first generation online comparison-shopping agent in the United States was BargainFinder, which came out in 1995. It was at that time an online experiment to measure the reactions of consumers and merchants to price comparison provided by Web-based IDSSs. Another comparison-shopping agent to be released the same year was PriceWatch, although PriceWatch was engaging in a more serious online business. (Wan 2006, 155-157.) A good example of a modern day implementation of a comparison-shopping approach is the Amazon online shopping website with their product comparison as well as recommendation tools.

3.6.2 Military applications

Another application of IDSSs is in the field of military decision making. From national defense budget planning to situation assessment, IDSSs have played an invariably important role in the direction of creating sophisticated military DSSs. Wen and Wang and Wang (2004, 55) researched on a KD-DSSs to support The Republic of China's national security budget planning based on the fact that there was a need to manage properly the change in national defense expenditures and satisfy many organizations who are affected by the change. The plan for this KD-DSSs is to "provide and refresh real-time information which decision-making officers can use to keep abreast of the implementation status of relevant policies so as to achieve the goals prescribed by the budgetary plan. The IDSSs for national defense budget planning aims to effectively make decisions on the processes involved in budget planning". (Wen et al. 2004, 55-56.) Military situation assessment is one of the processes involved in military decision making where the aim is to create relevant relations between objects in the environment. IDSSs that support effective and timely situation assessment and mission selection would be utmost important to military decision makers. (Xiang & Zeng & Zhu & Poh 2008, 158.) To build the KD-DSSs suited for situation assessment, the system needs an expert knowledge base that can deal with "uncertain and incomplete information and dynamically responds to the changes in environment." More specifically, this system is to have a model construction system, an interface for information input and decision model display output, and interface that handles maintenance and knowledge input. (Xiang et al. 2008, 159.) A visual representation of this system is in figure 4.

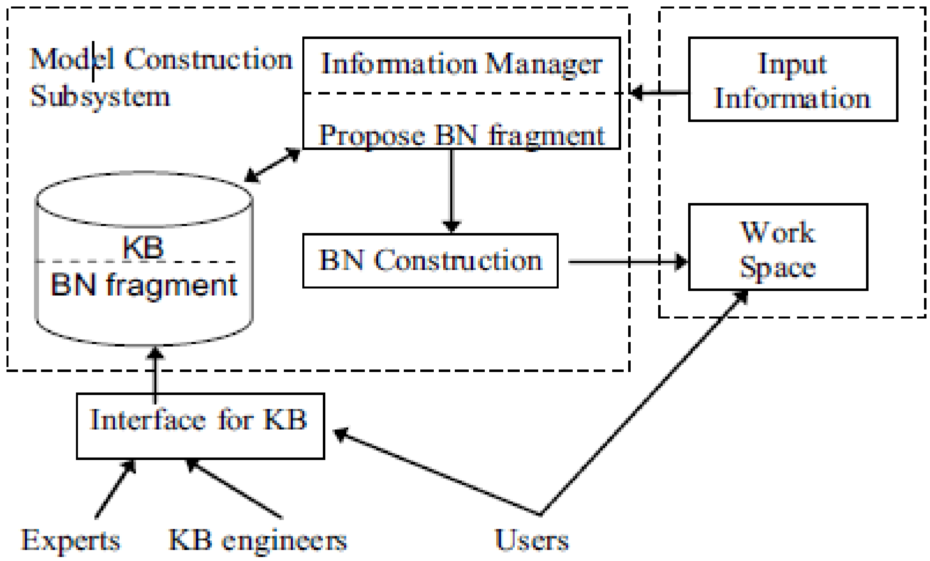

Figure 4. System architecture (Xiang et al. 2008, 159)

The Integrated Marine Multi-Agent Command and Control System (IMMACCS) is another example of a military DSS that was field tested in 1999 during the Urban Warrior Advanced Warfighting Experiment held by the Marine Corps Warfighting Laboratory in Monterey and Oakland, California, USA. The IMMACCS is a multi-agent, adaptive, distributed and open architecture system "that is intended to assist military commanders in making decisions under battle-like conditions when dynamic information changes, complex relationships, and time pressures tend to stress the cognitive capabilities of decision makers and their staff." (Pohl & Wood & Pohl & Chapman 2001, 1.)

Figure 5 displays the visual representation of the three-tier architecture of IMMACCS.

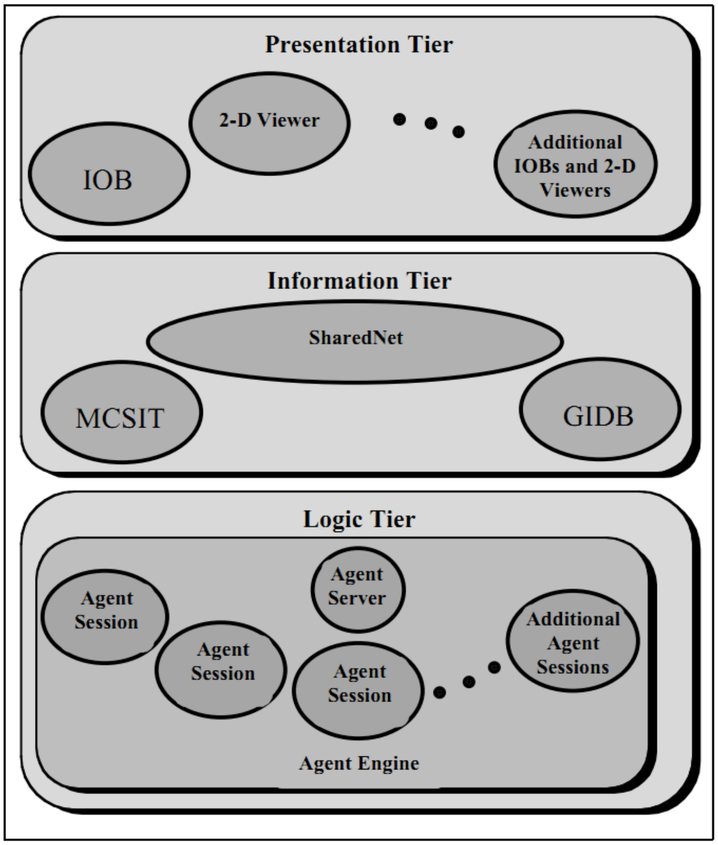

Figure 5. IMMACCS's Three-tier architecture (Pohl et al. 2001, 5)

IMMACCS incorporates a three-tier architecture that clearly separates presentation, information and logic. The Shared Net is a library of objectified information through which clients can retrieve and submit knowledge (Pohl et al. 2001, 5). The concept of objectified information is semantically based as it involves representing computer information "as objects with behavioral characteristics and relationships to other objects" (Myers et al. 1993 as cited by Pohl et al. 2001, 3). The agent engine in IMMACCS represents the logic-tier, consisting of an agent server that can serve many agents working as a group. This group is self-contained, self-managing and capable of both sending and retrieving information from the Shared Net. This group of agents work together by collection objectified information as the battlespace changes and interacting with each other and with the Shared Net. (Pohl et al. 2001, 6.) Lastly, the presentation tier that sits on top of the information tier and the logic tier consists of the 2D-Viewer and the IMMACCS

Object Browser, collectively known as the Client User Interface (CUI). Designed to give the user robust and graphical representation of information and analysis provided by the tiers below, the CUI can also be a collaborative tool in which decision makers can share tasks and strategies within fellow members who are also using CUIs. (Pohl et al. 2001, 6-7.)

3.6.3 Industrial applications

IDSSs have also found their way into production/manufacturing environments where managers are constantly exposed to problems and opportunities while at the same time have to make decisions which are both effective and efficient. This is why managers of manufacturing systems need DSSs which incorporate intelligence, or to be clearer, DSSs that are capable of supporting decision makers throughout the decision-making life cycle. (Delen & Pratt 2006, 325.) Managers are expected to make correct decisions in a timely manner, and their actions may result in changing the manufacturing system, while the model builder updates the IDSSs based on the characteristics retrieved from the manufacturing system. The Intelligent Manufacturing DSS itself plays the vital role of bridging the gap between the manager and the manufacturing system. Figure 6 shows the decision-making lifecycle in manufacturing systems.

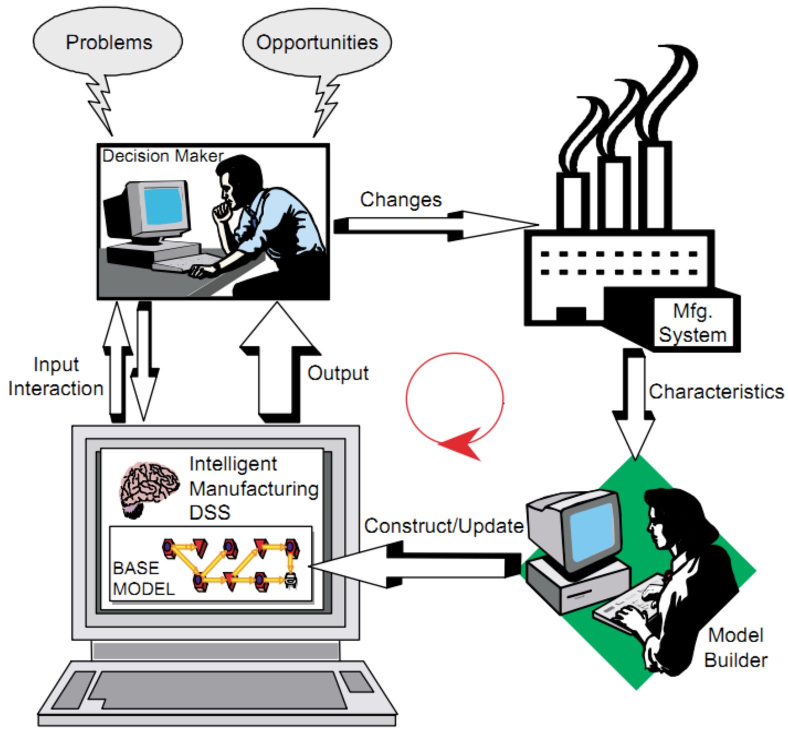

Figure 6. Manufacturing systems decision making life-cycle (Delen & Pratt 2006, 326)

Delen & Pratt (2006, 326) had designed, developed and prototyped a software environment for an intelligent decision support system for manufacturing systems (IDSS-MSs) that is capable of helping the manager throughout the decision making life-cycle, "which includes (1) structuring a problem from a given set of symptoms, (2) once the problems are structured, determining the best analysis tool (i.e. model type) to address the problem, (3) automatically generating the executable models specific to the structured problem, (4) conducting the analysis, and (5) providing the results back to the decision maker in an easily understandable format."

The software architecture of the IDSS-MS is shown in figure 7. This architecture is "an extended version of arguably the most commonly referenced architecture framework in the DSS literature, which was first introduced by Sprague (1980)" (Delen & Pratt 2006, 327).

The architecture of the IDSS-MS contains a data management sub-system, model-management sub-system, knowledge base sub-system and the user interface.

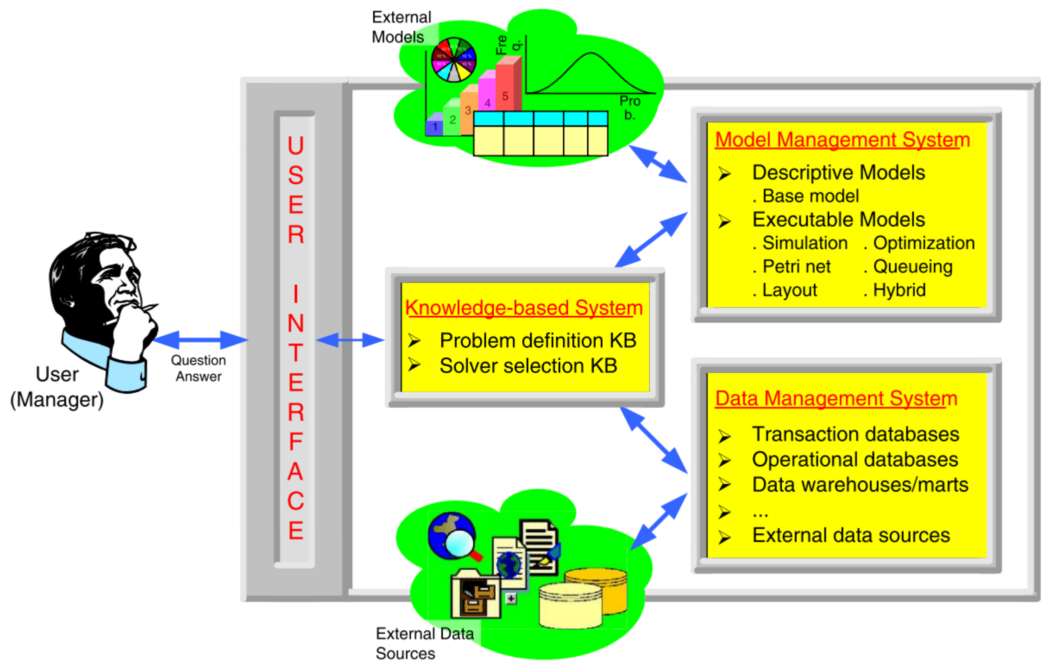

Figure 7. Conceptual systems architecture for IDSS-MSs (Delen & Pratt 2006, 328)

The data management sub-system connects to both the internal and external data repositories, providing accurate data/information to other system components when requested. The data management sub-system is implemented using an object-based mechanism supported by the connections to the relational databases (Delen & Pratt 2006, 327).

The model management sub-system of the IDSS-MS utilizes two major modeling concepts, the base model concept and the tool independent model representation concept, to supersede the traditional single-purpose, throw-away effort of traditional modeling methodologies (Delen & Pratt 2006, 327-328).

Firstly, the base model concept involves using a *base model* “which is an abstraction of a real world manufacturing system in the richest possible way (Duse et al. 1993 as cited by Delen & Pratt 2006, 328). The base model uses a library of manufacturing modeling primitives (programmed as general-purpose software objects) that are classified into three main types (Pratt et al. 1995 as cited by Delen & Pratt 2006, 328): physical primitives (machines, computers, parts etc.), information primitives (schedules, timelines, processes, bill of materials etc.), and control primitives (logic for choosing which part to process next, logic for determining which material handler to use etc.). Using this library, a model builder can easily build the manufacturing system specific base model by selecting appropriate modeling primitives and assembling them using a Windows driven software. The base model evolves with the organization and is persistent over time, and it is maintained and updated just as regularly as the company databases. Thus, the traditional way of modeling with a specific purpose in mind using a specific tool has given way to this new type of modeling. (Delen & Pratt 2006, 328-329.)

Second in the two major modeling concept that the IDSS-MS utilizes is the tool independent modeling concept, which helps managers derive the executable model from the description of the enterprise (in the form of base model). To apply this concept, the knowledge required for deriving the executable model from the base model must be encoded into business rules, integrated into the decision support system and be automatic and transparent to the user. (Delen & Pratt 2006, 329.)

Preventive Maintenance (PM) is the scheduling of maintenance tasks in attempt to prevent breakdown and failures before they occur. PM actions typically include such tasks as equipment check-ups, oil change of machinery, part replacement etc. Using PM, companies can expect long-term benefits including improved system reliability, decreased cost of replacement, decreased system downtime and better spare inventory management. Although it is true in some cases that PM tasks can be more costly than to repair the equipment when it is absolutely necessary, PM’s long-term benefits can prevent such hindrances as loss of production time and at the same time incur savings due to an increase of effective system service life. (Preventive Maintenance 2007.)

An example of an DSS that applies PM in its system is Optram by Bentley Systems Incorporated. The challenge that Optram was brought up to face was that railways of today “face ever increasing pressure from customers and owners to improve safety, capacity, and reliability of the rail system – while controlling expenses and tightening budget”. Optram is a comprehensive DSS that enables proactive maintenance of railway assets. It does so by comparing and analyzing volumes of current and historic track and rail asset data, enabling track and structure experts to improve maintenance planning and strategies. (Bentley Optram 2009, 1.)

3.6.4 Navigation and control applications

DSSs are also deployed in navigation and control systems where the management and guidance of air and sea traffic is crucial. The Lincoln Laboratory (LL) of Massachusetts Institute of Technology (MIT) is having ongoing development research to support the Federal Aviation Administration of USA in Air Traffic Control (ATC). Today LL focuses on developing safety applications, decision support services, and air traffic management automation tools, which aims at supporting FFA's next generation Air Transportation System. Due to weather-related delays, LL is helping the FFA to develop an integrated DSS that would alleviate such problems. (Air Traffic Control 2011.)

4 THE SEMANTIC WEB

This thesis aims to tie the functionalities on which IDSSs and Semantic Web technologies can collaborate. To provide foundations for this collaboration, the concepts of the Semantic Web and its technologies are described in this chapter. First, a brief history of the Semantic Web and its vision are explained. Then, descriptions of its core technologies are given. A few noteworthy applications of the Semantic Web are pointed consequently. Finally, the need for the integration of IDSSs and Semantic Web technologies is discussed.

4.1 Semantic Web technologies

The Semantic Web, a term coined by Sir Tim Berners-Lee, has closely been developed with the World Wide Web since the Semantic Web's beginnings (Hitzler & Krötzsch & Rudolph 2009). It is envisioned to give well-defined meaning (semantics) to information on the current Web, which in turn tightens the bond between human and machine. This extension of the current Web lies on top of the existing Web without having to modify it. Not only for the Web, the Semantic Web hopes to create a universal medium for information exchange by giving semantics, in a manner understandable by machines to the content of resources. (Trastour et al. 2002 as cited by Bose & Sugumaran 2007, 2.)

Semantic Web technologies enable Web developers to add defining tags to information in Web pages and add links to this information so that machines will be able to interpret this data, discover new data that is linked from this information and make associations between different data elements (Fensel & Hendler & Lieberman & Wahlster 2003 as cited by Bose & Sugumaran 2007, 2). The Semantic Web has three key technologies in reaching its objective: Extensible Markup Language (XML) for syntax and structure. Ontology systems that define terms and their relationships, and the Resource Description Framework (RDF) that provides a model for encoding the meaning defined through ontologies (Bose & Sugumaran 2007, 3). These three technologies are discussed below respectively.

XML is a popular and fundamental markup language for the World Wide Web. An example of a markup language is the Hyper Text Markup Language (HTML) which is used to describe elements of Web pages so that Web browsers can visually present them. XML is recommended by the World Wide Web Consortium (W3C) for data exchange and electronic publishing. (Hitzler et al. 2009, 353.) XML is not a programming language because similar to HTML, it does not actually do anything. XML uses elements and attributes to store data and encode information. Semantic Web technologies of today use XML for the serialization – which means the transformation of complex data structures into linear strings- of RDF content as essentially all programming languages have libraries which support the processing of XML files. This enables application developers to build on existing solutions for storage and processing. (Hitzler et al. 2009, 25-27.)

Ontology refers to a hierarchical data structure containing the relevant entities and their relationships and rules within a specific domain. Ontologies define data that are categorized in classes, subclasses and properties. The aim of building ontologies is to share and reuse knowledge. Since the Semantic Web is a distributed network, there can be dispersed ontologies that share semantically equivalent information. It is then important to map elements of these ontologies when processing information across applications or domains. (Bose & Sugumaran 2007, 5.) Ontologies enable a degree of semantic interoperability, which requires formal and explicit specifications of domain models that define the terms used and their relationships (Maeche & Staab 2001 as cited by Bose & Sugumaran 2007, 5).

The Web Ontology Language (OWL) is the W3C recommended standard for the modeling of ontologies since 2004, and many application domains have had increased usage of OWL ever since. OWL was chosen because it is an expressive representation language that allows developers to do logical reasoning based on knowledge, thus enabling access to knowledge which is only implicitly modeled. (Hitzler et al. 2009, 111.) OWL facilitates greater machine interoperability of Web content that supported by XML, RDF or RDF Schema by providing additional vocabulary along with formal semantics. OWL components include classes, properties and individuals. A Class is a concept in a domain, and it is also the basic building blocks of OWL ontologies. Classes have hierarchies such as subclasses and superclasses. Properties are split into two types: Object properties which relate one individual (instance of a class) to another and Datatype properties which relate individuals to datatypes such as integers and floats. (Bose & Sugumaran 2007, 5.)

The Resource Definition Framework (RDF) is formal language for describing information. It is often considered to be the basic representation format for developing the Semantic Web. This is because, similarly to HML and XML, RDF does not intend to display documents but allow them for further processing and recombination of the information that is within them. The goal of RDF is to enable applications to exchange information on the Web while preserve their meaning. (Hitzler et al. 2009, 19.) RDF makes use of Universal Resource Identifiers (URIs) to describe data, or resources, which could be people, places, documents etc. RDF provides a consistent, standardized way to describe and query Internet resources. (Bose & Sugumaran 2007, 4.)

In addition to RDF, The Resource Definition Framework Schema (RDFS) provides metadata for RDF vocabularies in the form of classes which itself is described using RDF. RDF is used to relate sources using properties while RDFS brings in the concepts of classes and hierarchies. Together, these two are termed RDF(S) and they form an ontology language for conceptual modeling with basic inference capabilities. (Grimm & Hitzler & Abecker 2010, 83.)

4.2 Applications of the Semantic Web

Applications that use Semantic Web technologies can be thought of as applications that apply technologies that relate to concepts such as metadata, data exchange and integration, knowledge representation, vocabularies and ontologies. It cannot be assumed that Semantic Web technologies are a concrete set of tools but a more vaguely defined class of tools as listed above because Semantic Web technologies are still too young for any formal definition. (Hitzler et al. 2009, 335.) Below are some examples of applications of Semantic Web technologies.

Metadata, or sometimes RDF, is integrated into many prominent websites and Web portals today, and they have become part of the linked data “cloud”. This “cloud” is a collection of data sets which have been interlinked by metadata, and these data sets include TV program information, music databases, prominent Web 2.0 portals such as MySpace and Flickr etc. A prominent example is DBPedia which is knowledge base that extracts structured data from Wikipedia. Freebase, part of the linked data cloud, is an open RDF-based database whose content integrates with data from the Web and can be edited freely by anyone. (Hitzler et al. 2009, 337.)

A vocabulary is a collection of identifiers with predefined meanings which are informally specified. In Sematic Web context vocabularies may have simple ontological relationships, and are occasionally used to transmit data between software applications. The RDF Site Summary 1.0 (RSS) is arguably the most commonly used Semantic Web vocabulary that is used in Weblogs (Blogs) and is expressed in RDF. Using an RDF feed reader, users can get updated content from websites whom provide these RDF feeds. The content usually contains regularly updated material such as news items and brief reports. Friend of a Friend (FOAF) is a vocabulary about persons and for social networking. FOAF uses RDF and OWL to describe the vocabulary. There are currently about 1 million FOAF files on the Internet with most being automatically generated on Web portals, and some by means of using generating tools such as FOAF Creator and FOAF-a-matic. (Hitzler et al. 2009, 337-340.)

There have been substantial research activities in an attempt to integrate Semantic Web technologies for Life Sciences. The idea behind these activities is to use Semantic Web technologies in order to facilitate and make the best use of the ever expanding knowledge in Life Sciences e.g. by establishing ways to integrate data repositories and to make them available in a structured form. Some examples of ontologies that have been made for Life Sciences include: the Gene Ontology which includes several of the world’s most important genome ontologies, the SNOMED CT (the Systematized Nomenclature of Medicine - Clinical Terms) ontology for clinical terminology designed to support the exchange and aggregation of health data and the GALLEN ontology which is designed for building clinical applications to support clinicians in day-to-day work. (Hitzler et al. 2009, 345-346.)

Semantic Web Services is a concept which involves using Semantic Web technologies to give machine-interpretable descriptions of current Web Services to enable better automation in Web Service searching and interoperability. Web Services is a type of middleware that supports Business-to-Business application interoperability, and is seen as a building block for the SOA approach (Fischer & Werner 2010). Semantic Web Services was the subject of research with aspects such as intelligent service discovery and fully automated service composition (Studer & Grimm & Abecker 2010, 6). Semantic Web Services was also researched in the field of distributed computing with such areas as Peer-to-Peer or Grid computing, which have seen Web Services as a base component for their underlying technology and to help them develop into Semantic Peer-to-Peer or Semantic Grid computing (Haase et al. 2004 as cited by Studer et al. 2010, 6; Polleres et al.2005 as cited by Studer et al. 2010, 6). Semantic Web Services is an important application of Semantic Web technologies as it is a possible direction in the pursuit of integrating Semantic Web technologies and IDSS. In the next chapter, a detailed description of Semantic Web Services and their potential integration with IDSS will be pointed out.

It can be seen from this section that even though the development of Semantic Web technologies is still ongoing, there have been applications that relate to and even use specific Semantic Web technologies in order to show the Semantic Web's ability to connect dispersed knowledge into one large shared collection.

4.3 The need for future research

With the prospective future of Semantic Web technologies being deployed in Web systems, IDSSs need reliable sources of knowledge in order to analyze intelligently and efficiently information and provide decision support capabilities. This thesis follows the works of a few notable researchers on their attempts of integration by drawing implications of their works. Below are a few reasons why IDSSs can and should harness the power of Semantic Web technologies to further enhance its decision-making ability.

"DSSs in the Internet Age will not be individual, interactive systems with specialized modeling features and interfaces, but will instead be combinations of data and services linked over the web for a specific purpose". These DSSs will be modular and reusable. The data from disparate domains which these DSSs use will be added with value in the form of expressive metadata models. These models will then be utilized by intelligent agents in order to support the decision making process. To achieve the above, successful DSSs must adapt to the challenges and opportunities of Internet-run world. (Hendler 1999 as cited by Casey & Austin 2002, 454.)

As the dynamics of the global market increase, the need for decision makers to have more diverse yet accurate and immediate information will remain rising. Also, the growth of the size of databases and human-centered information available to decision-makers expresses the need to represent this information in software-process able formats. At the same time, the availability of information from diverse sources on the Internet gives IDSSs more choices of input if this information can be made accessible by automated means. (Bose & Sugumaran 2007, 223.) Semantic Web technologies are made to support information-centric interoperability and reduce the cost of data reuse and provide intelligent content to be used by automated systems, enabling software services to perform distributed reasoning using Web resources (Dell'Erba 2004 as cited by Bose & Sugumaran 2007, 223). Within this environment, intelligent agents will be able to solve complex problems by gathering information from diverse sources and then combining the results that are in line with the decision makers' set parameters and criteria (Benjamins 2002; Hendler 2001 as cited by Bose & Sugumaran 2007, 223).

5 SEMANTIC WEB SERVICES

One of the potential directions of Semantic Web applications is the development of Semantic Web Services. Web Services are based on the Service-Oriented Architecture (SOA) approach in the modern software industry by enabling web-based, modular implementation of complex, distributed software systems. The ultimate goal of SOA aided by Web Services is to facilitate modular, interoperable, large-scale software systems because these systems provide "a *standard* architecture for modular systems, for creating new functionality from existing building blocks, and for enabling communication between heterogeneous elements". (Studer et al. 2010, 1.)

This chapter is dedicated to Web Services and the vision of Semantic Web Services fueling future web-based IDSS applications. This vision lies not on any concrete belief that such technologies being discussed will become prevalent in the near future. Instead, this vision can be thought of more as a proposition or a plan for the future. The first section of this chapter presents a definition of Web Services and their role in a SOA world. The second section then gives insight into the possibilities of Semantic Web technologies supporting Web Services to create Semantic Web Services. The third section proposes a vision of a SOA world enhance by Semantic Web Services. The final section gives a proposition on how Semantic Web Services can be used to build future IDSS applications.

5.1 Web Services as a building block for SOA

Web Services have recently come about and evolved to fulfill the vision of a Service Oriented Architecture (SOA) approach in the field of Business-to-Business (B2B) and Enterprise Application Integration (EAI). This vision arose from the need for business applications to communicate and integrate with each other. (Fischer & Werner 2010.) B2B and EAI are technologies for distributed and dispersed businesses to interoperate, and they are dependent on middleware to support communication and integration. These middleware serve as a means for a software application to provide services to another between the boundaries of the applications and the network through which they connect. Existing middleware solutions for B2B or EAI applications include the Common Object Request Broker Architecture (CORBA), Oracle Fusion Middleware, Java's Database connectivity Application Programming Interface (JDBC), ONC RPC, etc… Although these middleware are still being used, Web Services have the technologies to take a dominant stand in the middleware category, and will continue to exceed existing middleware to be used as the primary technologies in the SOA approach (Fischer & Werner 2010, 19).

As pointed out by Fischer & Werner (2010, 19), Web Services tackle the shortcomings of existing middleware and shed light on what could be the future of a SOA world. Firstly, Web Services solve the complexity and lack of standardization of existing middleware by using open-source and globally recognized standards to run its core technologies. Using the Extensible Mark-up Language (XML) and Universal Resource Identifiers (URIs), the primary infrastructure for Web Services are defined. URIs or its common subset, the Uniform Resource Locator (URL), has massive infrastructure supporting the technology such as the Domain Name System (DNS). Therefore, Web Services can be benefited by existing technologies and there is no need of inventing new ones. The other core technology of Web Services, XML, has multiple purposes; firstly as a means of structuring data, secondly as a standard for exchanging messages and thirdly as a language for describing services. The Simple Object Access Protocol (SOAP) is the protocol for exchanging XML messages, and it is also written in XML. SOAP basically define how XML Web Service messages look like, and defines a few *message exchange patterns* that can be used by Web Service partners (Studer et al. 2010, 20). Other than SOAP, existing standards such as the web protocol HTTP or the email protocol SMTP works well, and thus Web Services builders can focus less on solving message exchange issues and more on ways to improve other technologies. For describing Web Service interfaces, the Web Service Description Language (WSDL) is available and is also written in XML. It describes how inbound and outbound messages look like and where the Web Service is located using URIs, so once a user knows of such descriptions, they are able to locate and use the service. The other shortcoming that Web Services pushed away was the lack of support by large software companies. Existing middleware such as Microsoft's DCOM has never been supported by Sun, nor has Sun's Java 2 Enterprise Edition (J2EE) by Microsoft. Web Services have got much support from many developers who created new languages, protocols and tools. Web Service tools are being integrated in large enterprise application architectures such as Microsoft's .NET and Sun's J2EE and applications have been built with components from these 2 architectures. (Fischer & Werner 2010, 17-21.)

Despite the obvious advantages of Web Services over existing middleware, Web Services have challenges before they can be accepted in the SOA world. The first challenge is to have reliable directory services that list the Web Services available on the Web so that they can be easily searched and used. The Universal Description, Discovery, and Integration (UDDI) registry used today does just that, but is still lacking aspects to make the Web Service architecture fluent and reliable. UDDI allows everyone to add and modify Web Service entries, and while this seems to allow a large collection of Web Services to be added, it significantly reduces the quality of the Web Service entries in its registry. Therefore, a quality-assurance method must be implemented for Web Service directories. The second challenge is to have a standard and consistent security measure for Web Services. Security measures such as XML encryption, SOAP proxies, and XML signature for Web Service message authentication exists today, including many more. The problem here is that there are no agreed upon security measures, and a small set of security standards must be implemented before Web Services is going to evolve any better. The last challenge is interestingly the issue of interoperability. The problem lies in that companies may change the Web Service standards to fit with their intentions and that there are different versions of these Web Service standards, making different versions of Web Services standards not interoperable with each other. The organization WS-Interoperability (WS-I) is attempting to mitigate this problem by introducing *profiles* –a collection of standards- which leads companies to use Web Services standards that are compatible with a certain profile. (Fischer & Werner 2010, 22-23.)

Figure 8 describes a SOA built on top of the existing Web. Web Services standards and technologies including WSDL, UDDI registry and SOAP help to create an interface to connect service consumers to the services they need. Also this diagram shows the roles and relationships of each of these technologies and standards in making Web Services possible, with URIs and XML being the means to facilitate these relationships.

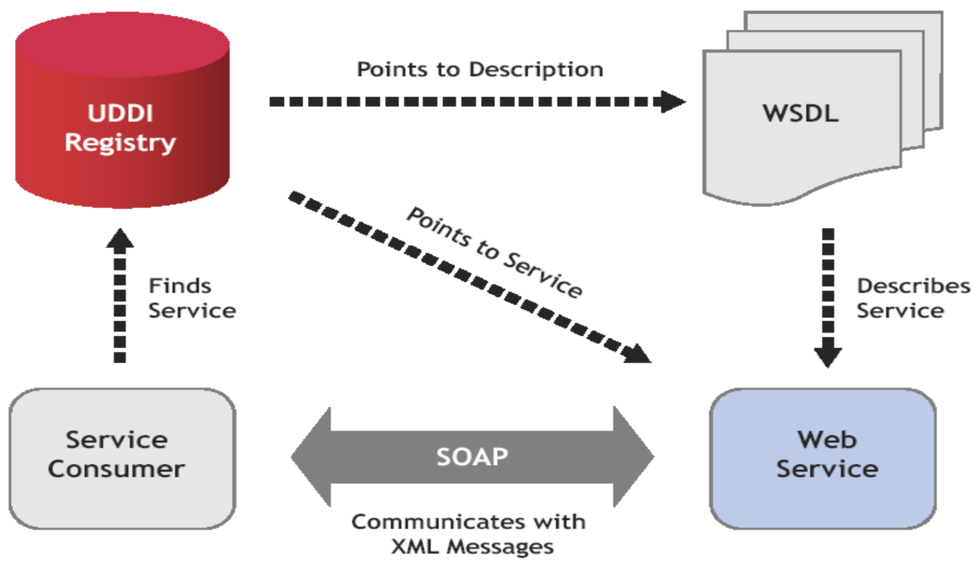

Figure 8. Web-based SOA as a new system design paradigm (Semantic Web Services Tutorial 2005, 13)

There are ways to overcome the challenges described above, and apart from the solutions mentioned, adding semantics to current Web Service standards and technologies is seemingly the next important step in making Web Services even better. The next section discusses why the Semantic Web research path should coincide with that of Web Services, and later sections will describe how a SOA Web could enhance Web-based DSS more effectively.

5.2 Adding semantics to Web Services

What if the potential of Web Services as a core technology for SOA can be elevated even higher than what it has already achieved? If the vision of the Semantic Web as to given machine-interpretable representations of information on the Web is to happen in the near future, Semantic Web technologies would help to address the shortcomings of Web Services and to enhance its core technologies even more.

The Semantic Web was brought up as a concept for turning information on the Web into machine-process able format. Meaning is given to data so that machines would be able to understand the data to a certain extent and human users would require less interaction with machines. Semantic Web technologies include RDF, RDFS, ontologies and XML, and these technologies combined help to realize the many potential applications of Semantic Web such as Semantic Search, Semantic Web portals, Semantic Wikis and of course Web Services. What Web Services currently lack is a machine-interpretable description of what the software system giving its services does, or what sequence of messages is used to interact with it. Currently UDDI does not have the technologies to enable intelligent searching of Web Services. A software system of a business can describe its services using ontologies to enable a more sophisticated discovery of the service. When semantically defined, a Web Service description can be reached by more intelligent ways, and thus are open to more service consumers. Also, having itself described in ontology will enable Web searching and advertising services to discover the Web Service much easier than before. A sense of automation can be achieved by adding Semantics to Web Services: Companies can find Web Services that are relevant to their needs automatically due to this intelligent discovery of Web Services, and even a complex service can be formed by automatically combining several smaller services. (Preist 2010, 159-160.)

Semantic Web offers tools which can extend its benefits to more specialized areas such as DSS which is the focus of this research. Semantic Web and Web Services combined, called Semantic Web Services, bring potential benefits to DSSs. The importance of Semantic Web Services in a DSS environment is discussed in the next two sections.

5.3 The vision of a semantically-enriched SOA world

In a world where SOA is the new means for system design and software development, Semantic Web Services would be a promising candidate to make this world function. Web Services serve as an interface for organizations to link their software services to the world via the Web, and in an SOA world, this is how software should be engineered. In fact, software is not necessarily built up by code; they can simply be a collection of different services acquired through the Web to perform a specific task. The function of software or application is, after all, to perform a task. Thus, this vision of a SOA future may redefine what people would think about an application, as they are no longer downloaded and installed, but the process of searching and using an application is a much more transparent process and requires less interaction from the user. (Ryabov 2012.)

To illustrate how the usage of software can become more transparent and simplified, let's take a scenario where a Web user needs to convert an image file from a JPEG format to a BMP format, and assume that the user is in a world where SOA and the Semantic Web are fully implemented. The user would simply need to type in a query such as *convert jpeg to bmp* and the Web interface, instead of showing link results like in traditional Web search engines, would analyze the query of the user, search for required Web Service, and finally have the user take command of the Web Service from there. This process is transparent in that the user does not need to know how the machine finds the Web Services, nor do they need to manually configure the service for their need. IAs on the Web have analyzed the user's query and searched for the appropriate tools using semantically enriched information linked by ontologies on the Web. This process is simplified in that users do not need to make the hassle of searching, installing and configuring themselves. All of the required steps are done by the machine, thanks to the help of semantics.

Individual needs are satisfied, but on the organizational level, the Semantic Web and SOA would bring great benefits to businesses as well. In a perfect SOA world an application in need of a service would query a directory service in order to receive the appropriate services that have been automatically searched for and matched with the requirements of that application. The best matches would then be automatically bound to the running application. In a more specific case, if two companies form a new strategic alliance or a new customer-supplier relationship in the real world, they can simply abandon unwanted or outdated services and select new ones. (Fischer & Werner 2010, 23-24.)

5.4 DSSs in a semantically-enriched SOA world

When Semantic Web Services govern the SOA world, many fields will benefit from the advantages that it gives. Studer et al. (2010, 6-7) analyzed and collected some published applications from different fields in which Semantic Web Services have been deployed, although most are in their prototype status. Some of these fields include logistics, tourism, finance, telecommunication, and business. The field of DSSs in general and IDSSs in particular is a natural candidate to employ Semantic Web Services in its design, implementation and functionality. As described in Chapter 3, IDSSs are component based, and such modularity would seem fitting to the aims of the SOA approach. This sub-section describes how IDSSs are going to fully benefit when deployed in a semantically-enriched SOA world.

The SOA approach promotes the building of applications that enhances flexibility and raises the level of abstraction for developers. In the vision of this approach, applications are no longer programmed but rather fitted together from different components, modules, or classes, each with their own functionalities and properties. These components that form a complete application are retrieved from the Web as services, and they are used dynamically so that they are ready to be searched for and used whenever needed. (Fischer & Werner 2010, 23.) Currently, Web-based IDSSs components communicate using Internet technologies and are connected on the Web through local or external servers (Turban et al. 2011, 85-87). If deployed in a SOA world with Semantic Web Services as the primary application component facilitator, an IDSS could be built from these modular Web Services with more automation and intelligence than before. Web Services that have been defined by knowledge representation languages and referred to by linked ontologies would give customers who search for these services a highly accurate result. Thus, developers could tailor the IDSS according to the requirements set by decision makers, enabling flexibility and control over the decision making process. IDSSs within this SOA environment could also have intelligent modules that automatically search and combine Semantic Web Services and give results to the decision maker, making the decision process to be seamless and transparent.

The SOA approach in building IDSSs can be further expressed in an example. In this example, an IDSS within a SOA environment using Semantic Web Services needs to be implemented in a governmental institute that needs to make a decision on how to overcome traffic congestion problems in a large city. By helping decision makers throughout the decision making process, the IDSS will automate the steps needed to reach the final decision.

In the intelligence phase, the IDSS will attempt to define the problem such as defining the causes: traffic congestion problems may be caused by the state of current infrastructure, the increase in vehicles, the traffic laws currently being imposed, and the increase in population. These causes are intelligently deduced with the help of ontologies and knowledge-representation languages that have predefined and connected these problems and causes in a logical manner. These ontologies may lie in the open Web or within the governmental institute in which the system is located. Also during this phase, the system will retrieve such information through Web Services. These Web Services are provided by other governmental organizations or publically on the Web, and with the help of Semantic Web technologies the searching and matching process becomes transparent as the system is able to understand the problem and analyze required information.

In the design phase, the IDSS will query the Web for Web Services which are able to construct models from the gathered information in the previous phase. For this scenario, the models can be graphs depicting the current state of infrastructure and the annual cost in building them, or the amount of vehicles in the city within a period of time. Models, together with the help of the user interface component of the IDSS, help decision makers visualize the information gathered and hide the complexities of the problem being analyzed.

After the models are shown to decision makers along with a list of choices comes the next phase in the decision making process, the choice phase. Here the IDSS may include software, or again request the functionality from Web Services, to simulate various scenarios of the choices, such as what-if scenarios and goal-seek scenarios. The what-if scenarios in this example would be raising and lowering the investment to fix rundown infrastructure, adjusting various traffic laws and lowering the limit of vehicles allowed in the city. An example of a goal-seek scenario is to present the question "To prevent traffic congestion in the central business district, what parameters need to be adjusted?" to the system and have it present all possible alternatives. An IDSS in this Web-based environment will typically have the user interface module through a web browser, and the standards in computer graphics and hardware would have been significantly higher in this proposed future, thus enabling easier understanding of the models and hiding the complexity of the decision-making process. Finally, should the decision makers not decide to let the system take the recommended choice by itself, they can simply revise the models and make the final decision by themselves.

As can be seen from the example above, IDSSs deployed on the Web are accessed through normal client computers running a Web interface that connects various components and/or Web Services together. With the help of Semantic Web technologies, the decision making process can be simplified and made increasingly efficient as the system can understand the problem better and assist the decision makers in more ways than before. The next chapter in this thesis will go into detail of the integration of IDSSs and Semantic Web technologies, and will give more specific implications of this integration.

6 INTELLIGENT DECISION SUPPORT SYSTEMS USING SEMANTIC WEB TECHNOLOGIES

In the previous chapters, studies into the background of these two research fields have given insight and suggestions to their integration. This chapter sheds light on Semantic Web technologies can be integrated in IDSSs. The first section analyzes previous research by presenting their findings and drawing out similarities to this work. The second and third section proposes a future of the Semantic Web and a SOA world with Semantic Web Services. The final section presents a conceptual model of an IDSS environment using Semantic Web Services and Semantic ontologies within the proposed future.

6.1 Related research

Several research works focused on integrating DSSs with Semantic Web technologies (Lu et al. 2006; Bose & Sugumaran 2007; Kwon 2003). Although these research works share the same overall objective of integrating regular DSSs with new technologies, the level of the integration defers between them. That is, Semantic Web technologies may be adapted directly into components of DSSs or act as a separate, external component. These research works also defer in the extent of their application, which consequently divides this section into two subsections: practically applying Semantic Web technologies in current DSSs solutions and proposing a framework for the integration.

6.1.1 Practical approaches

Lu et al. (2006, 321) proposed ways to enhance Railway DSSs using Semantic Web technologies. The need for their research arose from the fact that in the UK, the privatized railway network has caused various technical issues within the network to be separated in terms of responsibility. They conjectured that an intelligent, Web-based railway DSS is needed to help with the management and performance of the vehicle and track system.

Lu et al. (2006, 321-322) based their belief on several reasons that stemmed from the needs and requirements from the railway industry. In the railway domain, the vehicle designer must take into account the effects that changes to the vehicle speed, design parameters of the wheels and rails, manufacturing processes etc. have on vehicle derailment, ride performance and wear (Stribersky et al. 2004 as cited by Lu et al. 2006, 321-322). The railway network operator needs to balance the network performance against maintenance costs. The railway systems engineer needs to know the effect one component has on the others. The demand for DSSs to select the appropriate tool or a set of tools from over two hundred vehicles and track analysis tools (Bayati et al. 2002 as cited by Lu et al. 2006, 322) is apparent. With these needs being realized, a large comprehensive knowledge base, component of the proposed railway DSS, was needed to be created for rail vehicle and track system integration. This knowledge base was to serve railway personnel such as the railway designer, manufacturer, maintainer and operator by being able to query standards, patents and experts.

Semantic Web technologies were chosen to satisfy the needs explained above because of several reasons. Semantic Web ontologies are used to develop a framework for the creation of knowledge bases. Semantic Web technologies give the Web well-defined and interlinked information so that software agents can identify, interpret, manipulate, and interoperate the marked information among themselves (Hendler et al 2002 as cited by Lu et al. 2006, 322). Semantic Web ontologies, being the focus of Lu et al.'s research, have been implemented in the areas of intelligent reasoning, formation discovery, decision support, data fusion, systems integration and evolution of human knowledge (Rubin et al. 2004; Currie and Parmadee 2004; Flynn and Dean 2002a; Kogut and Heflin 2003; Berners-Lee et al. 2001 as cited by Lu et al. 2006, 322). The Semantic Web ontology has become a core in developing frame-based knowledge bases and subsequent applications.

Lu et al. (2006, 321) had developed two Semantic Web ontologies using Protégé. Protégé is an open source and W3C recommended ontology and knowledge-based editor that can support OWL. These two ontologies had been used to demonstrate their ability to infer new knowledge, help visualize and improve data presentation, query data and allow global database linkage as part of the Semantic Web. These ontologies were designed to be processed by intelligent, Web-based software agents. To create the ontologies, a framework was proposed to identify a large number of knowledge items for railway vehicle/track integration and to organize them in an object-oriented manner in terms of structural hierarchy (Lu et al. 2006, 330). The first ontology developed is the Railway Ontology for Vehicle and Track System that demonstrates features such as inference, visualization, and querying. The second ontology is to show a complete knowledge of UK mandatory standards of rail vehicles in order to help stakeholders in the UK railway industry manage the large amounts of railway standards.

It can be seen that Semantic Web ontologies have been realized and used as a solution to Lu et al.'s research problem. These ontologies can later on be as a part of the Semantic Web by being released publicly over the Internet. Consequently, existing IDSSs connected with the Semantic Web would interpret these ontologies and produce effective results.

6.1.2 Framework for future applications

The objective of this thesis ties closely to the work of Bose and Sugumaran (2007) as they propose a conceptual model of a Semantic DSS Environment Architecture. This research arose since the emergence of the Semantic Web and the goal was to make DSSs increasingly intelligent and efficient by integrating them with Semantic Web technologies. Bose and Sugumaran (2007, 227) defines their Semantic DSS Environment as an environment resulting from the combination of Semantic Web technologies and Web-based DSSs in an attempt "to bring out the best of each to create a more friendly, capable and intelligent DSSs environment for the user". By placing a strong emphasis on the potential of Web-based DSSs, they are able to draw out potential ways that the integration of the Web-based DSSs and Semantic Web technologies may work (Bose & Sugumaran 2007, 226).

The Semantic DSS Environment Architecture constitutes internal and external components. Internal components include an intelligent user interface, database and model base, knowledge base, ontologies, DSS manager, semantic search mechanisms, and Semantic Web Service discovery and composition mechanism. The external components include Semantic Web Service provides, Semantic Web Service registry, and ontological and knowledge resources. (Bose & Sugumaran 2007, 231-232.)

The architecture design does not stray far from the design explained in section 3.4, but despite this the design includes a few custom components that comprise the internal part of the DSS. First, the design includes an intelligent user interface supported by intelligent agents to help streamline the process of decision making and make interaction with the DSS more transparent. The next specialized component is the DSS Manager whose purpose is to link the data, model, knowledge, and user interface component of the DSS together and supports “what-if” scenario analyses. The final specialized component is the Semantic Search Mechanism component which includes the search of external information and ontological sources and the discovery, selection, composition, and quality monitoring operations of Semantic Web Services. (Bose & Sugumaran 2007, 232-234.)

The external components the Semantic DSS Environment Architecture include three important resources. The first source is the Semantic Web Service Providers which are mostly vendors that provide Web Services that may be used in the DSS. The second source is the Semantic Web Service Registries which are places where Web Service listings are located. The listings are described using ontology languages to help improve the advertising of the Web Services. The third source is the ontological and knowledge resources that hold information about web services, computational and model execution, and knowledge about the domain in which the DSS is specialized. (Bose & Sugumaran 2007, 235.)

In short, the Semantic DSS Environment Architecture has benefits such as being modular and scalable, adding customization support for DSS developers. This architecture also provides access to external sources which assists decision makers in dealing with problems that rely on information that is beyond the organization in which the DSS is located. The

Semantic Web ontologies help with the interoperability of the system and also improve communication with other existing applications (Bose & Sugumaran 2007, 235).

Another research to drawn from for this work is that of Kwon (2003) in which he focused on implementing Web-based open DSSs using Web Services. Kwon (2003, 378) developed a Web Service called the Meta Web Service that can handle federating Web Services. This development resulted in a prototype called the Research Web Service, which is a Meta Web Service designed and implemented using open DSSs concepts (Kwon 2003, 386). The framework of the Meta Web Service presents several concepts that are included in the design of the conceptual model in section 6.4.

The Meta Web Service framework includes three main components: the service finder, the service planner, and the service executor. The service finder receives the problem description from a user and uses ontologies on the Web to help understand the problem. Ontologies for the use of the Meta Web Service are described using DARPA Agent Markup Language (DAML) series markup languages. The service finder includes a Case Based Reasoning (CBR) capability to search existing plans quickly instead of having to make up a plan from scratch. These plans are past cases which have recorded successful usage of specific Web Services and the results of their usage. The service planner then gets the chosen case from the service finder. If the case is similar enough to be used in the current situation then it will be directly executed, if not the service planner will analyze Web Service profiles within a service registry to compose a new plan. Finally the user is presented with an application that is the solution to the initial problem. (Kwon 2003, 378.)

The framework of the Meta Web Service allows for a seamless service as the user does not need to know how the Web Services are retrieved and interconnected (Kwon 2003, 379). This automated Web Service execution serves as an important feature of an IDSS environment using Semantic Web technologies. How this automation will be carried out and the importance of which are explained in the section 6.4. The next section proposes a future environment that would potentially support IDSSs.

6.2 A prospective future with readily available Semantic Web technologies

The Web of today is the platform of choice for DSSs. The emergence of the Web has brought DSSs to new light, and Web-based DSSs are increasingly being researched to develop and implemented in areas such as health care, private companies, government, and education. The Web offers services for decision support, such as Online Analytical Processing (OLAP), data mining, data visualization, ad hoc query and reporting tools, and optimization and simulation models. It also provides the means for online marketplaces to exist so that DSSs can be built and deployed through the Web. (Bhargava & Power & Sun 2005, 1083-1085.) The question now is how can current practices of the Web of today be improved in the prospective future of the Semantic Web.

If the Semantic Web is to become the next evolution of the current Web, Semantic Web applications such as Semantic Wikis, Semantic Portals, Semantic Metadata, or Semantic Web Services would replace their Web 2.0 counterparts. These applications are but some of possible information sources DSSs of the future could harness from. For example, the portals and wikis of the Semantic Web future are given machine-interpretable content, thus can be automatically and intelligently retrieved by DSSs without any human help. These semantically-enriched websites of the future may also interconnect with each other and help information seekers find the closest match to their search. Semantic information portals that instruct how to design and implement DSSs would provide more relevant resources and give developers a more in-depth view of domain in which they will build the DSSs, all with the help of ontologies and semantic searching technologies. DSSs vendors would be able to advertise their applications more effectively and consequently have their products reach out to more potential customers. The possibility to build DSSs from Web resources through electronic retailing and Web Services have already seen its beginning in Web 2.0, and the Semantic Web with more interoperability and platform-independent features would likely to enhance the building of DSSs even further (Bhargava et al. 2005, 1088).

For the objectives of this thesis, not all Semantic Web applications are considered for the conceptual model. Semantic Web applications such as Semantic Wikis and Semantic Portals serve as references to building DSSs, and they may be good information sources during the intelligence phase of decision making. However, the envisioned model of an IDSS using Semantic Web technologies would harness the web-based, modular, and interoperable features that Semantic Web Services offer and supporting ontologies to enhance the intelligence of the system. The envisioned model is discussed in more detail in section 6.4.

In this prospective future, Semantic Web Services would be in effect and alter the way DSSs are built and distributed. Semantic Web Services would enhance traditional Web Services by enabling automatic dynamic interaction between software systems. Semantic Web Services technologies provide interfaces for businesses to deliver their software product as services in a much more intelligent and effective way. (Preist 2010, 159.) Data-driven and model-driven DSSs have differences in way they are distributed, with the latter being potentially more complex than the former. Thus model-driven DSSs are less inclined to be distributed through the Web (Bhargava et al. 2005, 1088). By enabling Semantic Web Services in the future, the organizations creating these complex systems can apply a new business approach by producing modular systems whose components can be distributed as a Web Service. DSSs of the future can then be built and used by combining multiple Semantic Web Services.

The Web has served as a platform for DSSs vendors to conduct advertising and commercial distribution of their products, and allowed users to perform such tasks as ordering and payment of DSSs products. To make Semantic Web Services work for DSSs, builders of DSSs must overcome possible economic challenges. If in the future DSSs are made from Semantic Web Services, then the services to make them will not come free, and DSSs vendors must come up with a clearly defined revenue model. An example of a working revenue model is salesforce.com that delivers CRM solutions to businesses through the Web. Currently, salesforce.com’s CRM solution includes such services as Sales Cloud, Service Cloud, Chatter, and Database.com. A look into the pricing plan of Service Cloud shows that it offers three service editions that are priced per user per month on an annual basis. This type of pricing scheme is relevant for Web-based DSSs because the stress on the servers and other resources provided by Web-based DSSs vendors depend on the number of users. (Bhargava et al. 2005, 1088-1092; Salesforce.com 2012.) Consequently, DSSs vendors must choose appropriate pricing plans that relate to their own resources and deployment methods, as well as their customer base.

In short, the prospective future of Semantic Web technologies enhancing IDSSs environment assumes a number of readily available technologies. Once these technologies are realized and applied, the creation of an IDSS environment using such technologies would be feasible. The next section classifies Web technologies currently available to build future Web-based IDSS applications.

6.3 Semantic Web Services and other Web technologies to support IDSSs

This section will categorize current Web technologies that may potentially support Web-based IDSSs. The understanding of available Web technologies in addition to Semantic Web possibilities may help in creating Web-based IDSSs easier and more sophisticated. Currently, Web technologies can be divided by those that enable "(1) server-side computation, (2) client-side computation, and (3) a distributed implementation and deployment of DSSs components (Bhargava & Krishnan 1998 as cited by Bhargava et al. 2005, 1086). Although defined more than a decade ago, this classification still holds true today and many Web technologies mostly fall into either one of these three classifications.

Server-side computation Web technologies provide platform-independent and universal access to decision support applications. These include Hypertext Preprocessor (PHP), Java Server Pages (JSP), Active Service Pages (ASP), and other server-side scripting languages. Client-side computation Web technologies are usually integrated with the user interface to extend its capabilities. Examples include Silverlight, Ajax, Java applets, Flash, and Flex. The last category includes technologies that offer a distributed implementation of deployment of DSS components. They include Enterprise Java Beans, COM+, Java RMI, CORBA, and Web Services. Web-based DSSs can be built using these technologies, and these Web technologies have changed the way DSSs are developed, deployed, and used. (Bhargava et al. 2005, 1086.)

As explained in section 4.1, Semantic Web technologies revolve around RDF, RDFS, OWL, and XML. There are ontology editors currently available to develop ontologies and support systems requiring knowledge management. In the Semantic Web future, these editors are likely to be developed even further and ontologies would be the main component for any semantically-enriched website or service. For example, there is an open-source application titled Protégé that is equipped with tools to develop ontologies and knowledge-bases. According to Protégé's main website, "Protégé is based on Java, is extensible, and provides a plug-and-play environment that makes it a flexible base for rapid prototyping and application development." Protégé supports ontology languages such as RDFS, OWL, and XML Schema. Supported by a strong community, Protégé have been used as a solution for knowledge management in such areas as biomedicine, intelligence gathering, and corporate modeling. (Protégé 2012.)

Should Semantic Web technologies become the norm of Web standards in the future, websites would be integrated seamlessly with semantic content, such as OWL and RDF. In fact, many of the creation of information sources such as wikis and portals could be deployed using Integrated Development Environments (IDEs) that transparently links these wikis and portals to ontologies on the Web. Not only serving the creation of websites, IDEs to develop components for IDSSs can also seamlessly integrate online ontologies and incorporate Semantic Web standards to these components. Many of the current programming languages such as C++ and Java would have extensive support for these Semantic Web technologies, thus providing fast deployment of semantically-enriched software.

The possibilities for creating intelligent software components using Semantic Web technologies are many. Through the previous chapter, Semantic Web Services is seen as a possible direction for future Web-based IDSSs development. If the future is as described in sections 5.3, 5.4, and 6.2, then the way Web-based IDSSs are designed, developed, and deployed would make a notable change. The following section describes in detail the change of direction to which the design of Web-based IDSSs can take in order to perform well in the proposed future.

6.4 A conceptual model of an IDSS environment using Semantic Web technologies

Given the prospect of widespread use of Semantic Web technologies as was described in sections 5.3, 5.4, and 6.2, a good foundation can be established to enable the creation of an IDSS environment that is built entirely using Semantic Web Services and dependent on ontologies on the Web to facilitate its intelligent components. This section describes such an environment, and is illustrated in a model created by using the vector graphics software Adobe Illustrator in a KTUAS learning facility. All of the images contained in the model are originally drawn.

Figure 9 below illustrates an IDSS environment using Semantic Web technologies (SIDSS).

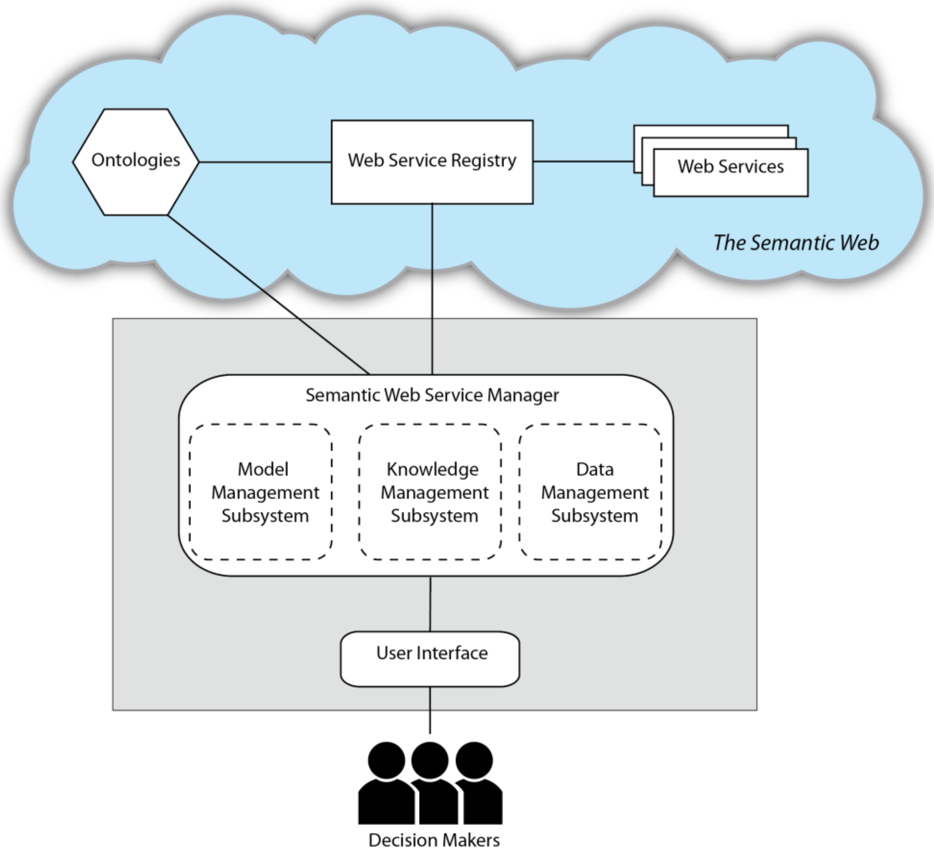

Figure 9. Web-based IDSS using Semantic Web technologies

The boundary of the system is covered by the grey rectangle, and the Web using Semantic Web technologies is represented by the blue cloud illustration. In this representation of the Semantic Web, ontologies serve as a knowledge base for use by the SIDSS and the Semantic Web Service registry. Semantic Web Services are intelligently described and machine processable, and are registered through the Semantic Web Service registry. In the SIDSS, the Semantic Web Service Manager (SWSM) acts as the component that is responsible for integrating and managing other components of the SIDSS. The user interface of the SIDSS controls the SWSM, and is run through a Web browser. The final component of the SIDSS is the decision makers who are using the system.

In the SIDSS, the SWSM combines the three subsystems of a traditional IDSS (see section 3.4 for more details on the components of a traditional IDSS). The SWSM in this scenario is made using server-side computation Web technologies. Either way, the SWSM is the bare bone component of the SIDSS environment, along with the User Interface. The SWSM's task is comparable to that of Bose & Sugumaran's (2007, 234) DSS Manager and Kwon's (2003, 378) Meta Web Service which are described in sub section 6.1.2. The SWSM handles the tasks of discovery, selection, composition, and quality monitoring of the Semantic Web Services.

Based on the decision making process shown in Ryabov's (2011) model of an IDSS, illustrated in figure 1 in section 3.3, the same process can be applied to the SIDSS through the interaction with Semantic Web Services. The first step in the SIDSS's decision making process is receiving input from the decision makers to define the requirements and parameters to a problem. These requirements and parameters are then analyzed by the SWSM with the help of online ontologies that the SWSM is connected to. The SWSM is supported by intelligent software that provides sufficient reasoning capabilities, preferably through AI techniques such as IAs, ANNs, and ESs. The SWSM then queries the Web Service Registry to find the Web Services whose descriptions match the requirements of the analyzed requirements and parameters. The Web Service Registry also uses online ontologies to assist it in the requirements matching process. The selected Web Services then fill the roles of the SIDSS's subsystems. In the model, the three subsystems are represented by dotted lines to show that they can be tailored according to specific needs. Thus, not in every decision making case all subsystems should be present. Through Web Service composition, the SWSM would make sure that the selected Web Services are compatible and work together seamlessly. The last task that the SWSM does is to monitor the performance during the execution of the selected Web Services. To ensure top quality performance, the SWSM may replace the installed Web Services with newer and more efficient ones, again through querying the Web Service Registry.

The User Interface of the SIDSS incorporates client-side computation Web technologies, and is able to directly manipulate the SWSM as well as any Web Services that are installed within it. Both the SWSM and the User Interface exist on servers on the Web, enabling platform-independent computing and limits the burden of maintaining hardware and software for an organization using the SIDSS.

In the SIDSS, there are no dedicated databases or model bases. Instead, the databases and model bases for use in an organization using the SIDSS would be selected by either the installed Semantic Web Services or by the SWSM. The organization may then lease these databases and model bases. Because of this reason, organizations using the SIDSS will need fewer resources to carry out their decision making, as well as any maintenance tasks done on the databases and model bases.

Due to the modular and SOA nature of Semantic Web Services, the service providers must provide working payment models in order to make the SIDSS feasible to implement. These Semantic Web Services can be leased based on a period of time, or based on the number of users, or both. Whatever the choices, the providers of these services must take into account that organizations will analyze their business needs and see whether or not it is more economical to obtain a decision support application through Semantic Web Services.

In overall, the SIDSS is envisioned to provide a responsive, transparent, and intelligent approach to computer-aided decision making. With Semantic Web Services as its building blocks, the SIDSS delivers platform-independent computing for any types of organization. The SIDSS runs on the Web, ensuring ease of access even in remote environments. The intelligent and automatic aspect of the SWSM reduces the workload of the decision makers: the decision makers only need to concentrate on defining the problem to the system and choosing an alternative based on the system's output. All processes in between are automatically handled by the SIDSS. The SIDSS may be feasible in the near future as the technologies to support it already exist, and continuing research on this subject may draw this future even closer. The implications of the SIDSS, as well as future research of the subject, are discussed more thoroughly in the final chapter.

7 CONCLUSION

IDSSs incorporate intelligence to assist decision makers by providing reasoning and knowledge capabilities. Integrating Semantic Web technologies, whose purpose is to provide machine-interpretable information, is a likely direction to the IDSSs research field. Although the definition of DSSs is not new, and the fundamental design principles of DSSs are persistent even until recent years, the rise of Web technologies has helped DSSs research to achieve more results. Semantic Web technologies, although not widespread, give rise to possibilities to make DSSs intelligent.

The integration of IDSSs and Semantic Web technologies is the focus of this thesis, and by studying the background of the two research fields, a solid foundation was made to facilitate the creation of a conceptual model illustrating the integration. This thesis presents and evaluates a number of different technologies and applications of IDSSs and Semantic Web technologies. Through this evaluation process Semantic Web Services, combining Semantic Web and Web Service technologies, were chosen to be the main technology to support the integration. Web ontologies using Semantic Web ontology languages also have an important role in the integration.

The choice of using Semantic Web Services has some limitations. Being deployed on the Web, Web-based IDSSs are faced with several technological challenges. Being stateless in nature, the HTTP protocol does not provide persistent connections and state. Decision support applications require repeated interactions with the models and the exchange of large amounts of data across multiple interactions. The Web in its current state will need to be upgraded in reliability and speed to reduce delays, and present alternative technologies to enable persistent operations. (Bhargava et al 2005, 1091.) Also, even equipped with strong encryption algorithms, transacting and storing sensitive internal corporate data and models through the public Web is riskier than through a localized network. With the proposition of using Semantic Web Services to build future IDSSs, the modular, customizable, automatic, and transparent aspects of Semantic Web Services mean that these IDSSs are more open and available to the general public. This may reduce the dedicated and domain-specific values of DSSs to a certain extent, which means that Web-based IDSSs may lack certain abilities that only a custom made, specialized IDSS may possess. Therefore, Web-based IDSSs developers must design user interfaces as well as provide specialize services that conform to casual as well as professional users, depending on the need. Finally, Semantic Web Services that offer decision support computations will likely not be free, thus these service vendors must choose appropriate pricing plans in order to effectively sell their services.

Nevertheless, there are many benefits to Semantic Web technologies powering IDSSs. The first obvious advantage is that IDSSs can understand semantics, or the meaning of data and information, so that they can intelligently perform with limited intervention by human users. Also, data and models are kept on the Web and are dispersed, but with the help of Semantic Web ontologies, these data and model sources are interoperated. Also, knowledge from different domains is combined according to these ontologies. With the help of Semantic Web Services, organizations running the IDSSs are released from the burden of owning, operating, and maintaining the IDSSs as they are owned by third-party service providers. Lastly, Web-based IDSSs provide remote, platform-independent operations as these systems can be accessed by any Web browser conforming to latest standards.

While it is clear that the conceptual model proposed in this thesis faces some immediate limitations due to its Web-based nature, the majority of these limitations can be overcome with further research and technological advancements. Together with the benefits described above, the Web-based IDSS using Semantic Web technologies promise to deliver fast and high quality decision making output, and with time further research may increase the effectiveness and efficiency of the system.

Using Semantic Web Services is just one of the intelligent technologies IDSSs can be integrated with. Future researchers may look further into the subjects of Expert Systems, Artificial Neural Networks, Semantic Web ontologies, and intelligent agents. Regarding the continuation of the subject of this thesis, further research into Semantic Web Services can be made to exploit their advantages to IDSSs environments and even to other computing fields.

REFERENCES

Air Traffic Control 2011. Lincoln Laboratory, Massachusetts Institute of Technology. Downloaded in September 10th 2011. <http://www.ll.mit.edu/mission/aviation/aviationresearch.html>

Bentley Optram Railway Maintenance Decision Support System 2009. Product Family Brochure .Bentley Systems, Incorporated.

Bhargava, Hemant K., Power, Daniel J. & Sun, Daewon 2005. Progress in Web-based decision support technologies. Whinston, A.B. (ed.) Decision Support Systems 43(4). 1083-1095.

Bose, Ranjit & Sugumaran, Vijayan 2007. Semantic Web Technologies for Enhancing Intelligent DSS Environment. Decision Support for Global Enterprises. Springer. 221 – 238.

Casey, Michael J. & Austin, Mark A. 2002. Semantic Web Methodologies for Spatial Decision Support. Proceedings of DSIAge2002: International Conference for Decision Support Systems in the Internet Age. Cork, Ireland. 453- 463.

Crunk, John& North, Max M. 2007. Decision Support Systems and Artificial Intelligence Technologies in Aid of Information Systems Based Marketing. International Management Review Volume 3. 61-67.

De Koch, Erika 2003. Decentralizing the codification of rules in a decision support expert knowledge base. University of Pretoria.

Delen, Dursun & Pratt, David B. 2006. An integrated and intelligent DSS for manufacturing systems. In Liebowitz, Stan J. (ed.) Expert Systems with Applications 30(2). Elsevier Ltd. 325-336.

Foster, Darren & McGregor, Carolyn & El-Masri, Samir 2005. A Survey of Agent-Based Intelligent Decision Support Systems to Support Clinical Management and Research.

Fischer, Stefan & Werner, Christian 2010. Towards Service-Oriented Architectures. In Studer, Rudi & Grimm, Stephan & Abecker, Andreas (eds.) Semantic Web Services. Springer. 15-24.

Grimm, Stephan & Hitzler, Pascal & Abecker, Andreas 2010. Knowledge Representation and Ontologies: Logic, Ontologies and Semantic Web Languages. In Studer, Rudi & Grimm, Stephan & Abecker, Andreas (eds.) Semantic Web Services.Springer. 51-106.

Herman, Ivan 2009. What is being done today? Deutsche Telekom Workshop Darmstadt, Germany. Downloaded 01 February 2011.

Hitzler, Pascal & Krötzsch, Markus & Rudolph, Sebastian 2009. Foundations of Semantic Web Technologies. CRC Press.

Kiong, Siew Wai & Rahman, Abd. Latif B. Abdul & Zaiyadi, Mohd Fairuz & Aziz, Azwan Abd 2005. Expert System in Real World Applications. Downloaded in November 5th.

<http://www.generation5.org/content/2005/Expert_System.asp>

Kwon, O. Byung 2003. Meta web service: building web-based open decision support system based on web services. Elsevier Science Ltd. 375-389.

Lu, Jinwei & Roberts, Clive & Lang, Karl & Stirling, Alan & Madelin, Keith 2006. The Application of Semantic Web Technologies for Railway Decision Support. In Gupta, Jatinder N.D. & Forgionne & Guisseppi A. & Mora T., Manuel (eds.) Intelligent Decision-making Support Systems: Foundations, Applications and Challenges. Springer. 321 – 337.

Power, D.J. 2000. Supporting Decision-Makers: An expanded Framework. Downloaded in September 2nd 2011.

<http://dssresources.com/papers/supportingdm/sld001.htm>

Power, D.J 2007. A Brief History of Decision Support Systems. DSSResources.COM. Downloaded in September 2nd 2011.

< http://dssresources.com/history/dsshistory.html>

Power, Daniel J. 2009 .Decision Support Basics. Business Expert Press.

Michalewicz, Zbiegniew & Schmidt, Martin & Michalewicz, Matthew & Chiriac, Constantin 2005. Case study: An Intelligent Decision-Support System. In Hendler, James A. (ed.) IEEE Intelligent Systems Volume 20 (4). IEEE Computer Society. 44 – 49.

McDaniel Jr., Carl & Gates, Roger 2010. Marketing Research Essentials with SPSS. 7th Edition. John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ.

Pohl, Jens G. & Wood, Anthony A. & Pohl, Kym Jason & Chapman, Arthur J. 2001. IMMACCS: A Military Decision-Support System. Collaborative Agent Design Research Center.

Preist, Chris 2010. Goals and Vision: Combining Web Services with Semantic Web Technology. In Studer, Rudi & Grimm, Stephan & Abecker, Andreas (eds.) Semantic Web Services. Springer. 159-178.

Preventive Maintenance 2007. ReliaSoft Corporation. Downloaded in September 7th 2011 < http://www.weibull.com/SystemRelWeb/preventive_maintenance.htm>

Protégé 2012. Welcome to protégé. Stanford Center for Biomedical Informatics Research. Downloaded in March 10th 2012. < http://protege.stanford.edu>

Ryabov, Vladimir 2002. Handling Imperfect Temporal Relations, PhD dissertation, University of Jyväskylä.

Ryabov, Vladimir 2011. Principle Lecturer, Kemi-Tornio University of Applied Sciences, Tornio, 10.05.11.

Ryabov, Vladimir 2012. Principle Lecturer, Kemi-Tornio University of Applied Sciences, Tornio, 13.02.12.

Salesforce.com 2012. Service Cloud Pricing and Editions. Downloaded in March 10th 2012. <http://www.salesforce.com/eu/crm/editions-pricing-service.jsp?d=70130000000rzrT&internal=true>

Semantic Web Services Tutorial 2005. 2nd European Semantic Web Conference, Heraklion, Greece. Downloaded in February 20th 2012. < http://wsmo.org/TR/d17/resources/200505-ESWC05/SWStutorial-eswc05.ppt>

Semantics. (n.d.). Collins English Dictionary - Complete & Unabridged 10th Edition. Downloaded in November 10th, 2011. <http://dictionary.reference.com/browse/semantics>

Sim, Ida & Berlin, Amy 2003. A Framework for Classifying Decision Support Systems. AMIA Annual Symposium Proceedings. 599 – 603.

Studer, Rudi & Grimm, Stephan & Abecker, Andreas 2010. Introduction. In Studer, Rudi & Grimm, Stephan & Abecker, Andreas (eds.) Semantic Web Services. Springer. 1-11.

Turban, Efraim & Sharda, Ramesh & Delen, Dursun 2011. Decision Support and Business Intelligence Systems, Ninth Edition. Pearson Education Inc., Prentice Hall New Jersey.

Wan, Yun 2006. A New Paradigm for Developing Intelligent Decision Making Support Systems (iDMSS): A Case Study on the Development of Comparison Shopping Agents. In Gupta, J. & Forgionne, G, & Mora, M. (eds.) Intelligent Decision-Making Support Systems (iDMSS): Foundations, Applications and Challenges. Springer-Verlag London: London. 147-165.

Wang, Huaiqing 1997. Intelligent Agent-Assisted Decision Support Systems: Integration of Knowledge Discovery, Knowledge Analysis, and Group Decision Support. In Liebowitz, Stan J. (ed.) Expert Systems with Applications 12(3). Elsevier Science Ltd, Great Britain. 323-335.

Wen, Wu & Wang, Wei-Kang & Wang, C.H. 2004. A knowledge-based intelligent decision support system for national defense budget planning. Department of Accounting, YuanZe University, Taiwan, ROC.

Xiang, Yanping & Zeng, Yifeng & Zhu, William & Poh, Kim-Leng 2008. An intelligent design system for military situation assessment. International Conference in Intelligent System and Knowledge Engineering.