International Journal of Information Technology and Management

Vol. VII, Issue No. X, November-2014, ISSN 2249-4510

A STUDY ON DEVELOPING THE NEXT GENERATION OF DECISION SUPPORT SYSTEMS USING AUTONOMOUS SOFTWARE AGENTS

International Journal of Information Technology and Management
Vol. VII, Issue No. X, November-2014, ISSN 2249-4510

A Study on Developing the Next Generation of Decision Support Systems Using Autonomous Software Agents

Umesh Kumar Lilhore[1] Prof. Dr. Rishi Pal Bangarh[2]

[1]Research Scholar, Maharishi University of Information Technology, MUIT (Lucknow)

Abstract – IBM has predicted that software agents will become the most important computing paradigm in the next ten years (Gilbert, 2007). The growing number of commercial and research agent implementations provides evidence that the computing industry recognizes the potential of this new paradigm (Huhns & Singh, 2008; General Magic, 2008; Gilbert, 2007, Mitsubishi Electric, 2007; Object Space, 2007). The standard-setting bodies that have been formed to address the issues of agent communication and mobility furnish further confirmation (Chang & Lange, 2006; Neches, Fikes, Finin, Gruber, Patil, Senator & Swart out 2001).

---------------------------♦-----------------------------

INTRODUCTION

The DSS implementation described is based upon a case study used in a DSS textbook (Holsapple & Whinston, 2006) and supports variance-analysis investigation for a manufacturing firm. When a variance from the budget occurs in the *rate i.e., hourly cost* or *usage* of raw materials, labor or overhead, the DSS provides support for whether the variance should be investigated based upon criteria specified by the user and prior investigations of variances.

Upon accessing the DSS, an authorized user would select a product to analyze and would then view the current variances for that product. The standard and actual parameters are loaded from the database, with the DSS allowing the user to perform what-if analysis with these parameters, uploading any desired changes in budgeted parameters to the database. Variances that exceed a pre-determined cutoff value are flagged. Users can review and edit the cutoffs set for each parameter.

It is the general goal of this paper to explore the promise of software agents in the realm of Decision Support Systems (DSS) and to provide guidance for DSS developers seeking to agent-enable their applications. To do this we first note the current state of DSS development in general and then that of software agent implementations within DSS.

Additional criteria for investigating variances can be reviewed and edited. These criteria are used to assign a relative weight to the variances previously calculated and provide additional support for the user in deciding whether a variance should be investigated. Should the user so decide, the cost of the investigation and the sources of the variance revealed by the investigation can be recorded in the database and reviewed when future variances occur?

In the MBMS, modeling agents can generally be used to integrate and monitor the use of standalone applications, such as statistical and linear programming packages, as demonstrated in our agent-integrated DSS.

Modeling agents could also be used to implement modeling approaches that are not available in the stand-alone applications. These agents could utilize some form of machine learning, operations research methods, or other algorithms to produce decision-making alternatives. Due to the customizable nature of models, these autonomous agents will be difficult to reuse in entirety but large sections of the code should be extensible.

Meta-modeling agents could also be used in the MBMS to coordinate the development and selection of alternative solutions given the existence of multiple models within the DSS. These agents would furnish an alternative evaluation process that would provide support for DSS with multiple goals. Again, due to the customizable nature of modeling functions in DSS, the code reusability for these meta-modeling agents will be somewhat limited.

RESEARCH METHODOLOGY

Having studied the literature on non-DSS agent implementations, and having built an agent enabled

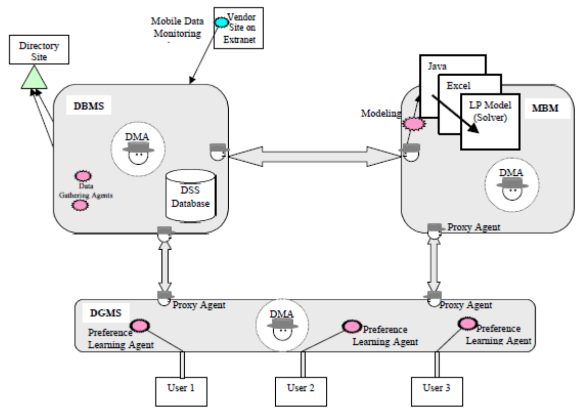
DSS, our efforts turned to developing a framework for an agent-enabled DSS.

Procedurally, we started by examining the Variance-Analysis DSS, and then attempted to generalize that architecture. Our philosophy was to build on fundamental approaches given in the literature, developing a first-cut framework that could be enhanced by others.

We started our framework with three fundamentals. The first fundamental, in keeping with Sprague and Carlson (2002), was to segment DSS into three components (DBMS, MBMS and MGMS). The second was that each DSS segment or component should be encapsulated, being kept as independent as possible. This principle was derived both from Sprague and Carlson themselves and from good programming practices. The third fundamental we adopted was to include in each DSS component an agent to oversee or manage the other agents within the component. We noted from the literature that it is common practice for a resource-manager agent to be used to monitor and control those agents performing common, functional tasks (Bradshaw et al., 2007).

The framework incorporating these fundamentals for the agent-enabled DSS we built is illustrated in Figure. Note several things with respect to that figure. First, the use of a (rounded) rectangular shape does not imply that any of the three DSS components contains only agents that are physically proximate.

Second, we moved toward encapsulation by incorporating *proxy agents*. These agents facilitate communication among the three domains. By insisting that information only flow between components via these conduits, a degree of independence and encapsulation of the domains is achieved. The proxy agents perform translation as necessary for information to flow.



Third, *domain manager agents,* one in each of the DSS components, take on the role of resource management and oversight of the other agents in their domain.

To extend the specific framework to a more general DSS framework, we reviewed and evaluated the agents we had built as well as others discussed in the literature. We concluded that all of the agents incorporated in Figure should remain in the general DSS and that others from the literature should be included as well.

Figure is the result of this review and is the general framework for an agent-enabled DSS. We now describe the remaining agents shown in Figure by DSS component.

In the DBMS the use of *data-gathering* and *data-monitoring* agents would be beneficial in DBMS of most DSS for gathering and maintaining information not typically stored in corporate databases. These two types of agents can be either *static (immobile) or mobile*, as needed. Due to the simplicity of both data-monitoring and data-gathering agents, the code for these agents is highly reusable, even across DSS of different domains.

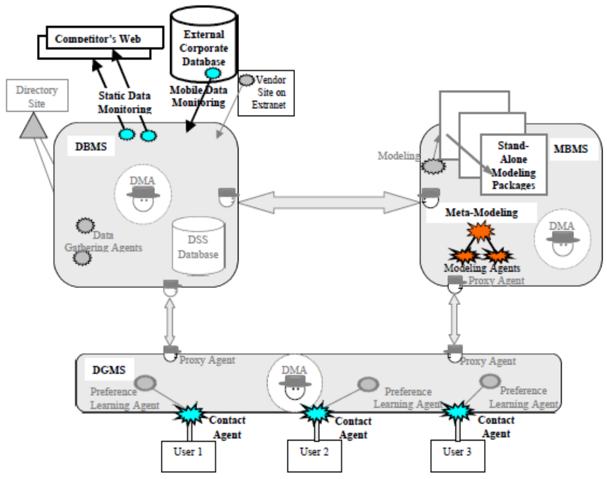
In the DGMS, p*reference-learning agents* could be created for each individual DSS user. These agents would be responsible for monitoring and storing the desired preferences of the assigned user, as suggested by the example preference-learning agent described in the Variance-Analysis DSS.

In addition, *contact agents* would be responsible for directly communicating with the user. These agents would notify the user of specific changes in the DSS environment (e.g., "the price has gone up") and would guide the user in efficiently utilizing the support provided by the DSS.

These general interface agents would work in a fashion similar to agents developed in the human computer interface stream (Erikson, 2007; Nardi, Miller, & Wright, 2008).

The general interface agents would be expected to be highly reusable, although special-purpose preferences would require considerable effort. For example, an agent that learns user-display preferences would be extensible, whereas a contact agent providing help screens would be highly system specific.

The framework shown in Figure is suggested as a starting point for DSS builders seeking to agent-enable their systems. The domain managers and proxy agents establish the basic building-blocks for integrating and managing agents within a DSS. The remaining agents provide examples of how agents can enhance the functionality of the DSS subsystems. Some aspects of the framework may not be appropriate for particular DSS and similarly, some DSS may benefit from agent uses that are not presented.



CONCLUSION

Decision making is the final goal of a decision support system (DSS), and, for environmental domains, it is multi-sectorial by nature. In practice, decisions can be taken by single authorities or by a group of responsible decision makers. Nowadays, decision makers use DSS, which serve as an informational background, enabling real-time simulation and further decision generation. Informational support, received with DSS, helps to mobilize and allocate resources, to set priorities and to share successful patterns and strategies in area of public health, air and water resources, and other. Nowadays, there are many definitions of what a DSS is. For example, DSS can be designed as a specific class of computerized information systems that support decision making activities. A properly designed DSS is an interactive software-based system intended to help decision makers in compiling useful information from raw data, documents, personal knowledge, and patterns to identify and solve problems and to make decisions.

REFERENCES

- Aggarwal, P. & Vaidyanathan, R. (2003). Eliciting online customers' preferences: conjoint vs. self-explicated attribute-level measurements. Journal of Marketing Management, 19, 157- 177. Amazon Homepage. (n.d.). Last accessed February 1, 2004.
- Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The semantic web. Scientific American, 284 (5), 34-43.
- Bhargava, H. & Power, D. (2001). Decision support systems and web technologies: a status report. Seventh Americas Conference on Information Systems (AMCIS '01). Association for Information Systems.
- Boston, MA. Bradshaw, J. M. (2007). Software Agents. Menlo Park: American Association for Artificial Intelligence.
- Bui, T. & Lee, J. (2009). An agent-based framework for building decision support systems. Decision Support Systems, 25(3), 225-237.
- Burke, L. (2001). Introduction to artificial neural systems for pattern regognition. Computers and Operations Research 18(2), 211-220.
- Buss, A. (2000). Component-based simulation modeling. Winter Simulation Conference (WSC '00). IEEE. Orlando, FL.
- Buss, A. & Stork, K. (2006). Discrete event simulation on the world wide web using java. Winter Simulation Conference (WSC '96). IEEE. Piscataway, NJ. Chu, P.C. & Beasley, J.E. (2007). A genetic algorithm for the generalized assignment problem. Computers and Operations Research 24(1), 17-23.
- Cohen, M., Kelly, C., & Medaglia, A. (2001). Decision support with web-enabled software. Interfaces, 31(2), 109-129.
- Cubert, R. & Fishwick, P. (2007). A framework for distributed object-oriented multimodeling and simulation, 2007 Winter Simulation Conference (WSC '97). IEEE. Atlanta, GA.