8-15-1997

Collaborative Environments for Leveraging Modeling Knowledge

Balasubramaniam P.
Boston University, bala@bu.edu

Melanie L. Lenard
Boston University, mlenard@bu.edu

Collaborative Environments for Leveraging Modeling Knowledge

P. Balasubramanian
Boston University, School of Management
595 Commonwealth Ave.
Boston, MA 02215
email: bala@bu.edu
Melanie L. Lenard
Boston University, School of Management
595 Commonwealth Ave.
Boston, MA 02215
email: mlenard@bu.edu

Abstract:

We explore how a collection of models and related modeling knowledge could be made available over an organizational Intranet. Because of their widespread use in organizations today, we plan to focus on spreadsheet models. We envision two approaches to a collaborative modeling environment: one is the "pull" model where knowledgeable users search for what they want, the other is the "push" model where the system distributes models and modeling knowledge to those who are likely to need them.

Introduction

Models have come to be recognized as important organizational resources. Work in the field of model management has tried to find ways to help decision makers make better use of models. However, most of the approaches in the literature deal with models that exist in one place. The recent growth of client-server applications and the World Wide Web has created new environments for applying the concept of model management systems.

Our intent in this paper is to explore how a collection of models and related modeling knowledge could be made available over an organizational Intranet. We believe that such a facility would not only promote sharing and re-use of models and but would also promote collaborative modeling work. In particular, we will focus on spreadsheet models as a vehicle for explaining and eventually demonstrating our concepts. We selected spreadsheets since it represents one of the most pervasive and successful application that has been developed for a personal computer.

Related Work

Bhargava et al [1] have developed DecisionNet to share models and solution algorithms over the Internet. For locating the models one needs, they propose two architectures each built for one of two modeling languages: AMPL and GAMS. Both architectures rely on model suppliers to register their product in a "yellow pages" that would be searched by potential model users. The architecture built for AMPL relies primarily on the user to be knowledgeable about models, methodologies, and languages. In contrast, the architecture built for GAMS expects very little modeling knowledge from the user. Instead, it requires model suppliers to provide more detailed information about the products they register and relies more on the system (in the form of "software agent") to help the user identify and use models.

In the area of spreadsheet modeling, Isakowitz et al [3] propose separating the logical and physical aspects of spreadsheet models in order to support the sharing and reuse of spreadsheet models in organizations. They have identified the primitives needed to model spreadsheets in addition to providing a factoring/synthesizing algorithm that will decompose/build spreadsheet models. They believe that their work would lead to the development of spreadsheet model management systems (SMMS).

Although, the ideas expounded in the above papers are very interesting, they need to be extended to deal with spreadsheets models over a corporate Intranet. In particular, we extend the spreadsheet modeling approach proposed in [3] by using a more general formalism (structured modeling). We believe that such an extension would be very useful, given that fact that spreadsheet modeling is, probably, the most used (and abused!) modeling technique in an organization.

Approaches

We envision two approaches to sharing modeling knowledge over an Intranet: one is the "pull" model, the other is the "push" model.

In the "pull" model, a potential user might begin by seeking information about models by searching the environment for "something useful". The least knowledgeable users might begin by looking for general guidance from a discussion group, perhaps being directed to the FAQ or other archival documents of the group. Armed with more knowledge, users would be able to specify more precisely what they needed and perform a search for models that are "close" matches to what they need. Eventually, they would narrow the search to a few of the most useful modeling modules, which they would retrieve and (try to) use. In trying to use the model, they might return to the discussion group or its archives for further guidance. (See Figure 1 for the architecture of the system.) Of course, to provide this functionality, the system should support storage, searching, and retrieval of models and related knowledge (stored with the models or captured in discussion groups). A critical element is the method(s) used for indexing [see, for example, Mannino, et al (1990)].

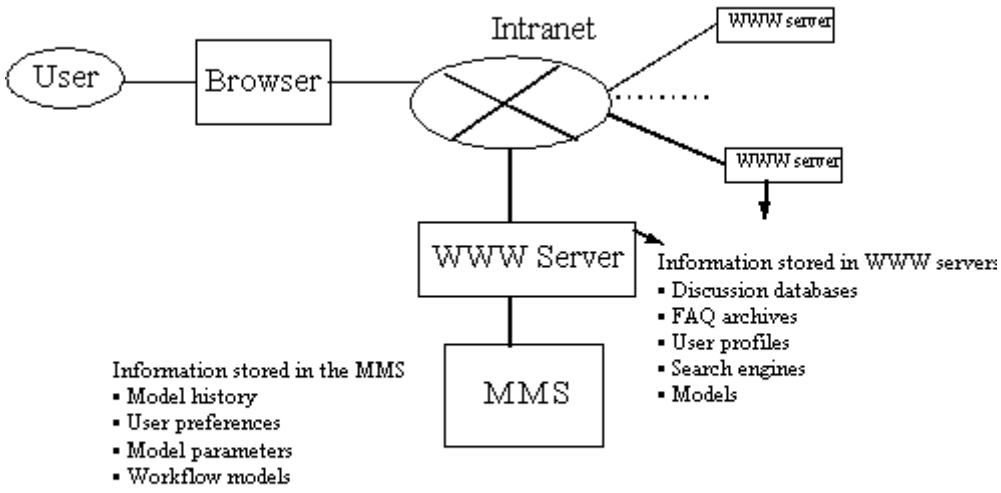

Figure 1: Architecture for a MMS Collaboratory

In the "push" model, the environment would somehow determine (with some help from the user) what models the user would want to know about and/or use and would send appropriate messages alerting him/her to their availability. Users might supply this information themselves, essentially by subscribing to a service, indicating their interests and preferences (e.g., text vs graphics, frequency of notification). However, if the environment contained organizational information about work flow [6], including the roles and responsibilities of each user, it might be able to deduce that Mary in Marketing will need a sales force scheduling model in the last week of every month. It might also be able, by recording Mary's use of

the environment, to learn that she responds best to weekly reminders and rejects models that require lots of data entry. If the systems keeps track of all the information generated and used during the model development life-cycle [4], the environment could also help her find or even supply the data needed to run the model and suggest where the results should be sent. Further, it could inform her of others who have used this model and might provide assistance. For delivering this functionality, the system should help users classify and register model sources, combine query languages, browsers and navigation, construct a plan to evaluate queries, etc. This functionality could be divided between the model management system and the server.

Future Work

The first decision we have to make will be about the modeling primitives for storing information about models (a meta-modeling language). For this, we plan to borrow ideas from structured modeling [2]. Structured modeling is a conceptual framework for representing models based on a set of interrelated definition of all the elements comprising a "model". This implies that it defines a "model of models" or a meta-model [7].

The next step will be to provide more details about the two architectures. In both cases (push and pull), the environment should assist all three categories of users: model builders, analysts and end-users. We plan to flesh this out in the future through an implementation using a collection of spreadsheet models.

Finally, we would like to describe our notion of such environments as providing a capability [5]. This means that in addition to the technical aspects of model sharing, while they need to be overcome, there are other hurdles to surmount. Issues such as how to encourage people to go through the effort of making their models available to others are likely to be much more difficult to deal with. This may call for new organizational forms and processes/methodologies to support the model management system.

Bibliography

[1] Bhargava, H., Krishnan, R., Roehrig, S., Casey, M., Kaplan, D. and Müller, R., Model Management in Electronic Markets for Decision Technologies: A Software Agent Approach. *Proceedings* of the 30th Hawaii International Conference on System Sciences, IEEE Computer Society Press, Maui, HI, January 1997.

[2] Geoffrion, A., An Introduction to Structured Modeling, *Management Science*, Vol. 33, No. 5, pp. 547-588, 1987.

[3] Isakowitz, T., Shocken, S. and Lucas, H., Toward a Logical/Physical Theory of Spreadsheet Modeling. *ACM Transactions on Information Systems*, Vol. 13, No. 1, pp. 1-37, January 1995.

[4] Krishnan, R., Model Management: Survey, Future Research Directions and a Bibliography. *ORSA CSTS Newsletter*, Vol. 14, No. 1, pp 7-16, Spring 1993.

[5] Kulatilaka, N., Balasubramanian, P, and Storck, J., Managing Information Technology Investments: A Capability-based Real Options Approach, Boston University, *Working Paper* #96-35, June 1996.

[6] Kwan, M. and Balasubramanian, P., Dynamic Workflow Management: A Framework for Modeling Workflows, *Proceedings* of the 30th Hawaii International Conference on System Sciences, IEEE Computer Society Press, Maui, HI, January 1997.

[7] Lenard, M., Fundamentals of Structured Modeling, *Mathematical Models for Decision Support*, Springer-Verlag Berlin Heidelberg, pp. 695-713, 1988.

[8] Mannino, M., Greenberg, B. and Neung Hong, S., Model Libraries: Knowledge Representation and Reasoning", *ORSA Journal on Computing*, Vol. 2, Nol 3, pp. 287-301, Summer 1990.