An Enterprise Modeling Approach to Organizational Decision Support

Sulin Ba, Andrew B. Whinston
Center for Information Systems Management
The University of Texas at Austin
Austin, TX 78712, U.S.A.

Karl R. Lang
Institut für Wirtschaftsinformatik und Operations Research
Freie Universität Berlin
Berlin, Germany

Abstract

A novel and comprehensive framework is outlined at a mostly conceptual level for future enterprise modeling systems. We discuss some major design issues and indicate the direction future research could take to support enterprise-wide problem formulation and problem solving. The focus of the paper is chiefly on investigating the question of how to better do model building, and how to extract the relevant parts of the model to support specific analyses of corporate issues in an enterprise-wide environment. Conceptually, we are looking for an enterprise modeling system (EMS) which automatically builds and executes task-specific models as needed in response to queries posed by the user. EMS is especially aimed at supporting strategic decision-making such as predicting the effects of changes in business policies, analyzing possible reactions to internal and external threats, and exploring new business opportunities.

1 Introduction

In order to respond to new challenges in an increasingly complex and dynamic environment, modern management is using a vast amount of knowledge from various sources. Depending on the particular problem being investigated, managers switch between different perspectives and level of details when searching for the relevant pieces of knowledge required to provide an appropriate answer. However, lacking a centralized knowledge management facility, individual managers' access to knowledge is restricted to a relatively small subset of the collective organizational knowledge, depending on their status and function within the organization, which inhibits the recognition of all interactions and interdependencies relevant to the problem under study. For better decision making, managers need to look at problems in a non-myopic fashion, and take on a global organizational view instead of individually biased perspectives.

Because many enterprise scenarios are too complex to be fully understood, models are developed to help decision-makers analyze specific situations. Models are constructed by choosing a particular view and by introducing assumptions, abstractions, and approximations. Together they create a simplified image of some segment of reality which permits us to study a given problem scenario in a systematic manner. The research on decision support systems (DSS) is concerned with the development and implementation of computer supported decision-making and problem-solving environments. Despite some twenty years of progress in DSS research and technology, current systems still lack the generality and versatility needed to supply managers with adequate DSS tools which indeed do support all phases of the enterprise modeling process.

Conceptually, we are looking for an enterprise modeling system (EMS)[1] which automatically builds and

[1] We use the term enterprise modeling in accordance with the definition provided in [20], p.19, where enterprise is defined as "a collection of business entities ... in functional symbiosis," and thus differs from the usage in the business re-engineering area. Business entities mean organizational (sub)units and (groups of) people, and functional symbiosis refers to the interactions among a set of intraorganizational as well as interorganizational entities sharing a common goal. Hence, the scope of enterprise modeling explicitly includes external partnerships like relationships of an organization with its suppliers, subcontractors, customers, and the public.

executes task-specific models as needed in response to queries posed by the user. EMS is especially aimed at supporting strategic decision-making such as predicting the effects of changes in business policies ("what if"-type of questions), analyzing possible reactions to internal and external threats ("what should we do"-type of questions), and exploring new business opportunities ("where should we go"-type of questions).

In our paper, we discuss some major design issues for the next generation of enterprise modeling systems, and indicate the direction future research could take to support enterprise-wide problem formulation and problem solving. The focus of the paper is chiefly on investigating the question of how to better do model building, and how to extract the relevant parts of the model to support specific analyses of corporate issues in an enterprise-wide environment. We discuss and propose ideas which we see as promising steps towards accomplishing this difficult endeavor. There are two, essentially disjoint, research efforts, one based in the artificial intelligence community ([1], [10], [21]) and the other in the decision support systems community ([3], [4], [5], [9], [13], [18]), which study model building and reasoning with multiple models. This paper draws upon both, and develops a synergistic framework for future enterprise modeling systems. Designing an organization-wide modeling system which satisfies the above requirements necessitates explicit reasoning about the model's underlying assumption and its scope of applicability.

We see organizational-wide reasoning systems as decision tools for both strategic[2] and operational management. We envision a system whose reasoning about a particular organization is based on a library of model fragments representing significant organizational phenomena and processes from different perspectives and at different level of details. Accomplishing this requires access to multiple sets of heterogeneous model fragments which differ in several dimensions, some of which might even be mutually inconsistent. We need to address the issue of model representation and model organization, that is, we need a language for expressing relationships of different kinds, and for expressing underlying assumptions controlling their applicability. The task of organizing organizational knowledge into semi-independent, reusable model fragments is a crucial one for enabling an EMS to compose useful, problem-specific models by integrating relevant, existing model components under a variety of different modeling circumstances.

[2] When exploring strategic questions, it is essential to be able to include qualitative knowledge into the analysis. While current DSS research emphasizes quantitative modeling, work in the area of qualitative reasoning has given focused attention to reasoning about qualitative knowledge.

2 Overview of the EMS

Researchers in the DSS and Artificial Intelligence (AI) communities have proposed several frameworks which provide partial solutions to this formidable problem. We argue that cross-fertilizing ideas from both research fields will achieve significant progress in answering many of the open research questions impeding the development of complete, enterprise-wide decision support systems. While the DSS and AI paradigms diverge in their application domains, management and engineering, respectively, they face basically the same underlying model building issues. Work in the two areas differ also in other aspects. Model management in the DSS field can be seen as a natural extension of previous work in management science and operations research, and has advanced mathematical modeling from a state where modeling was an uncoordinated task, whose success depended mainly on the technical skills and expertise of the user, to a state where systems actually know about certain types of mathematical models and appropriate solvers. Most of the DSS research in model integration has taken graph-based approaches. AI research, on the other hand, has addressed important issues, such as the explicit representation of modeling assumptions, and the usage and exploration of qualitative knowledge.

We propose an enterprise modeling system framework which draws upon ideas from both fields. Using different sets of assumptions and various kinds of knowledge, ranging from general, qualitative knowledge to specific and precise numerical models, the system helps managers analyze organizational questions from different perspectives and at different levels of detail. Given a particular task, model building is guided by the selection of an appropriate perspective and level of detail, a modeling decision for which little support is found in current decision support systems technology. The software architecture we propose for designing an EMS is depicted in figure 1.

The EMS is designed as an interactive software tool, which supports decision making and problem solving when exploring various business scenarios. It comprises five functional modules: The *query manager*, the *model manager*, the *candidate evaluation* module, the *solver*, and a *post processor*.

The query manager provides the interface between the user (e.g., an organizational decision maker) and the EMS. It processes a user's queries such as, for example, *"what is the effect on profits if we increase promotional expenditure of product line XY?"* and translates them into a set of executable statements which are submitted to the model manager, the core of the EMS. Model manager is the most sophisticated module which controls access to

Proceedings of the 28th Hawaii International Conference on System Sciences (HICSS '95)
1060-3425/95 $10.00 © 1995 IEEE

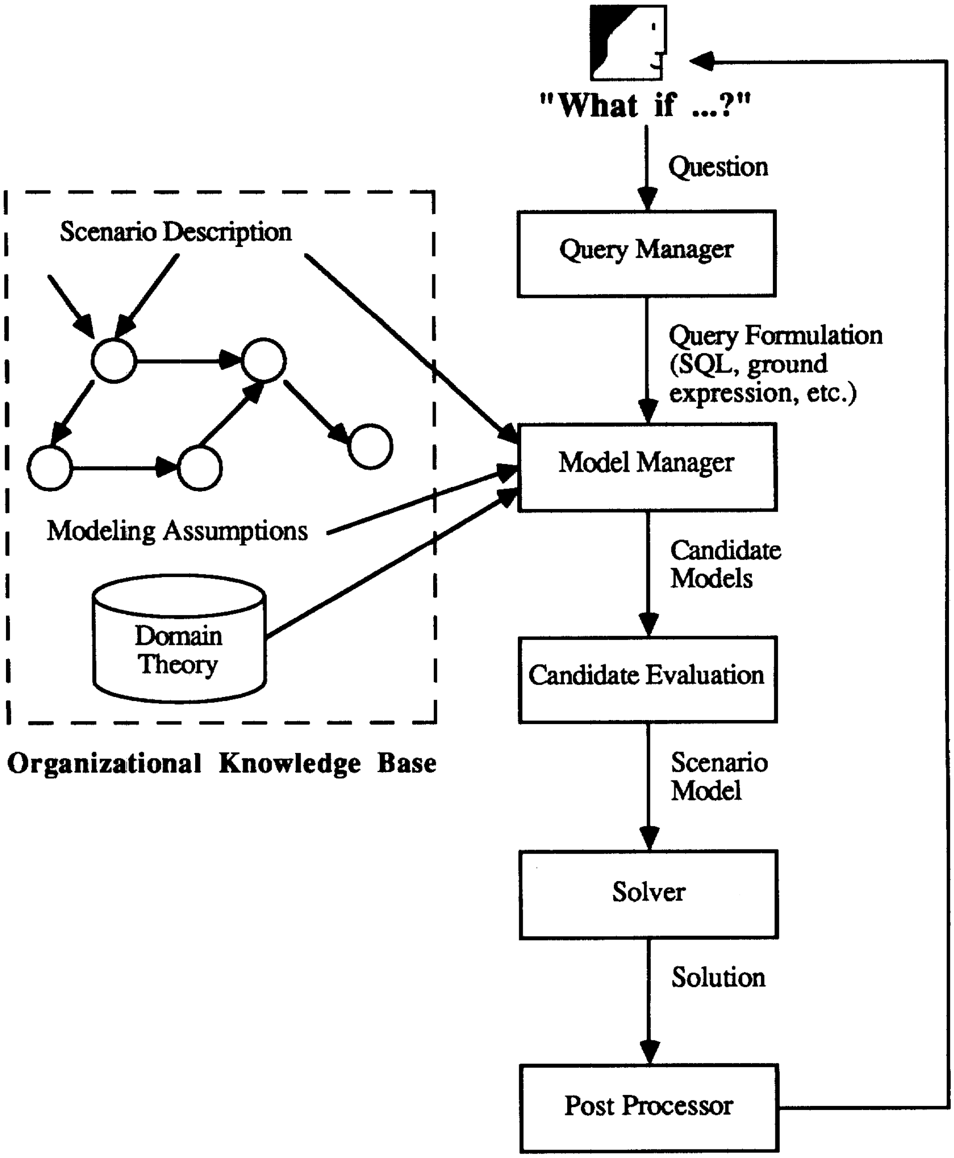


Figure 1: The software architecture for the EMS.

models and data in the organizational knowledge base. The enterprise modeling framework requires first the building of a general-purpose domain theory that describes a variety of organizational objects, activities, and processes, and additionally a collection of interesting scenario descriptions which reflects different views on the organization from different perspectives. The domain theory is represented as a library of model fragments, each describing an independent aspect from a particular viewpoint. It contains both qualitative[3] and quantitative relationships ranging from general organizational laws and rules to relationships that are very specific to a particular company. The explicit representation of modeling

[3] Researchers in artificial intelligence area ([8], [11], [16]) have done some important work in reasoning with qualitative knowledge. [12] and [13] have developed a framework and an algorithm for using qualitative knowledge to represent organizational relationships towards enterprise modeling.

assumptions in terms of abstraction level, approximation, perspective, and level of detail is another essential feature in enterprise modeling. Reasoning about those assumptions enables the EMS to identify a suitable collection of compatible model fragments and to build consistent, composite models in response to a query. Typically, there is no unique scenario model, and the model manager finds several feasible models, called candidate models, and passes each of them on to the next EMS module. The candidate evaluation module then collects all candidate models and chooses the best candidate as the final scenario model. In this context, best means most appropriate and useful, and refers to the simplest possible model that is coherent, comprehensive, and appropriate for the task. The solver module selects the adequate solution method and then solves or simulates the scenario model chosen by candidate evaluation module. Finally, a post processor is employed in order to translate the model solution into an intelligible answer which can be presented to the user in return to the original question.

3 Organization of the knowledge base

Similar to compositional modeling developed in artificial intelligence [10], we view the model base as a repository of organizational knowledge whose purpose is to provide a resource of sharable and reusable models for helping to better understand, explain, and predict organizational phenomena in a variety of different situations. In order to achieve the necessary depth and versatility, the model base needs to contain knowledge of different kinds which should include quantitative, qualitative, and hybrid forms of information.

The main challenge of designing such a knowledge base is to decompose the vast body of knowledge available into semi-independent fragments in a manner that allows us to assemble new, task-specific models under a wide range of scenarios. Merging relationships from multiple model fragments into an integrated, composite model requires not only a careful approach of grouping relationships into independently meaningful units, but also an explicit treatment of the modeling assumptions which describe when they apply.

The observation that a model consists of more than just a set of relationships, because it always assumes a particular modeling context, leads us to a definition of a model fragment where the modeling assumptions are explicitly and separately expressed from the actual relationships and constraints. Hence, we argue for different representation languages to represent the underlying assumptions of a model fragment and its constituting relationships. Each model fragment would contain two sections, one that contains the specification of modeling assumptions as predicates, and the other containing the actual constraints and relationships that apply if all predicates in the modeling assumptions section are true. The constraint section would specify a set of constraints and relationships that are imposed by the model fragment if its assumptions are validated. Before a compositional modeling algorithm can actually search the model base and identify task-specific, relevant model fragments, it needs sufficient information to be able to evaluate the predicates in the modeling assumptions section. This extra information needs to be derived from the query, or needs to be inferred from meta rules which would be specified as a part of the model base. These meta rules would rule out incoherent and inconsistent combinations of modeling assumptions, and would also imply additional conditions as a consequence of modeling assumptions that have been already established.

4 Representation of modeling assumptions and constraints

We distinguish between several types of modeling assumptions:

(1) Conditions providing structural and topological information on the organization considered. The entire organizational system should be organized into linked subsystems which can consist of other systems or primitive objects (individual variables). Part-of relations can be used to state, for example, that the manufacturing department is part of the company, and that plant X is part of manufacturing. Is-a relations can be used to express that widget 2000 is a new product, and that a new product is also, in a more general sense, just a product.

(2) Simplifying assumptions including those which shift focus to a particular perspective of the organization, and indicate if a cost analysis, a productivity analysis, or if some other kind of analysis is desired, or those that are mainly used to simplify a model for computational benefits, or those that are used to reduce the complexity of phenomena. For example, inventory models typically assume that new batches of products arrive at the beginning of a planning period, and are consumed at a constant, continuous rate, although product arrivals and departures are actually discrete events.

(3) Ontological assumptions which take a certain view on the organization, and select an appropriate method of description. Should the organization be viewed as a collection of employees who are working towards a common, cooperate goal? Should the organization be described as a collection of interacting subunits such as functional departments where the interaction might be represented as information flows, cash flows, or material flows?

Proceedings of the 28th Hawaii International Conference on System Sciences (HICSS '95)
1060-3425/95 $10.00 © 1995 IEEE

(4) Granularity assumptions determine the level of detail for a given analysis. A production scheduling analysis may require the consideration of each worker and piece of machinery involved in the manufacturing process of the products. A strategic marketing study might need a more aggregated view, and suggest a study in terms of product groups without explicitly considering any details of the manufacturing process.

(5) Operating assumptions help to focus model simulation by choosing, for example, if a static or a dynamic analysis is appropriate, or if a qualitative, a quantitative, or a hybrid form of model is desired. We think that these seven different types of modeling assumptions, which could be represented as first-order logical predicates, cover most distinctions which are implicitly made when modelers formulate traditional, monolithic models. As suggested by [6] and [10],when using first order predicate logic to represent the assumptions underlying a particular model component, one would specify each model component as a logical implication, where the set of the relationships would be the consequence, and the modeling assumptions, expressed as a conjunction of predicates, would be taken as the antecedent.

Different representational forms should be allowed to specify relationships or constraints. The abundance of uncertainties and vagueness, which are actually very characteristic of organizational knowledge, often inhibit the specification of precise quantitative models. Qualitative statements are typically based on hypothesized monotonic relationships of the form *if variable X is increased (or decreased) then variable Y will increase (or decrease)*. Those kinds of monotonically increasing (decreasing) relationships can be modeled using representations developed in qualitative reasoning field, such as in QSIM (Qualitative SIMulation) form as $y=M^+(x)$ and $y=M^-(x)$, respectively [16]. Furthermore, quantitative algebraic and differential equations, and interval-based hybrid forms (e.g., rules and constraints in Rules-Constraints-Reasoning (RCR) algorithm, see [15] and [13]) can also be used to integrate models that are formulated at different levels of detail. RCR models might also be useful for specifying purely qualitative model fragments, and incorporating numerical, company-specific information in the form of bounds in order to improve preciseness. In the following we discuss the specification of the domain theory.

5 Specification of the domain theory

The model formulation problem can be defined as selecting the relevant model fragments, and generating a composite, task-specific model which is coherent and most useful for answering a question asked by the user. More specifically,

Given

- a domain theory, consisting of a collection of model fragments describing certain organizational phenomena and a set of rules guiding how to use them;
- a scenario description, describing the logical structure of the organizational artifact being considered and a set of statements constraining its behavior;
- a query about the behavior of the scenario being considered,

Produce

- a composite, coherent and (most) useful scenario model which provides an answer to the posed query.

First of all, we need a domain theory describing the specific company under study. Constructing an exhaustive domain theory would be, of course, a tremendously time consuming and costly project in itself. Particular views on the organization are defined as scenario descriptions. Users would work with scenario descriptions suited for their particular task, and would issue queries with respect to a certain scenario.

In this paper, it shall suffice to merely sketch out such a domain theory. An elaborate example can be found in [2]. A model fragment definition consists of two major parts: one that contains the conditions under which the model fragment is applicable (called the modeling assumptions section), and the other that actually encodes the relationships of the model fragment (called the relations section). Domain model fragments are of the form

> **fragment** <NAME> (*output port*)
> {description of the functionality of the model fragment}
> **conditions**
> *precondition-specifications*
> **relations**
> *relationship-specifications*
> **end**

where <NAME> is an expression designating an instantiation of this model fragment, *output port* is a list of the variables which are computed by the model fragment, and which can be shared with other fragments. Variable names are written in upper case letters throughout the model definitions. The conditions section contains *precondition-specifications,* which define the modeling assumptions that an instantiation of a model fragment depends on.

The CORP Domain Theory listed below is a simple example of an organizational domain theory which consists of model fragments and meta rules. In each model fragment, the **consider** clause is used to describe certain perspectives, such as an organizational performance perspective, which we would specify as **consider(performance)**. Perspectives can be further focused, and a specification like **consider(performance(financial))** would imply a financial perspective on the organization. Granularity assumptions enforce a certain level of detail. The clause **exists(PRICE)**, for example, calls for a scenario model which has to include in the model the price of a particular product when reasoning about sales. Other modeling assumptions are specified in a similar way, like the operating assumption about the type of relationships (e.g., qualitative or quantitative) in the model , or the kind of analysis (e.g., dynamic or static) the model fragment could be used for. Finally, the relations section contains relationship specifications, which would be constraints of a particular modeling language. We only assume that internally, that is, within a single model fragment, the relationships are of a homogeneous type. Across model fragments, heterogeneous relationship specifications could co-exist by using several modeling languages.

CORP Domain Theory:

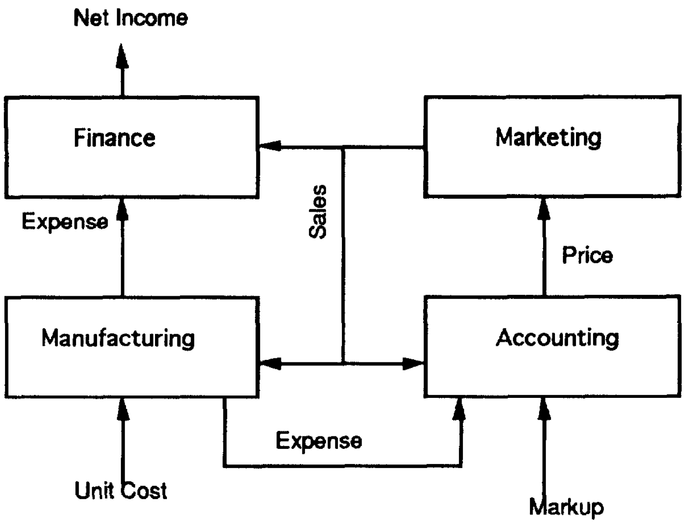
```
fragment MKT-1 (SALES)
    {marketing model describing qualitative
    relationship between price and sales volume}
conditions
    consider(exists(PRICE)),
    consider(performance(financial)),
    model-type(qual), simulation(dynamic)
relations
    SALES = M-(PRICE)
end

fragment MKT-2 (SALES)
    {marketing model describing quantitative
    relationship between price and sales volume}
conditions
    consider(exists(PRICE)),
    consider(performance(financial)),
    model-type(quant), simulation(static)
relations
    SALES = 80000 - 44000 * PRICE
end

fragment MKT-3 (SALES)
    {marketing model describing semi-qualitative
    relationship between price and sales volume}
conditions
    consider(exists(PRICE)),
    consider(performance(financial)),
    model-type(qual-quant), simulation(dynamic)
relations
    SALES(t)    = [68000,92000] -
    [40000,48000]*PRICE(t)
end

    .    .    .
    .    .    .
    .    .    .

meta rules
    R-1:  consider(performance(X)) and
          model-type(qual)=>simulation(dynamic).
    R-2:  model-type(qual) => solver(qsim)
end
```

Besides the definition of model fragment, a domain theory also contains meta rules which constrain the use of the model fragment, and thus help to eliminate potential model candidates. For example, it could be a reasonable assumption that qualitative performance studies always entail a dynamic analysis, a rule which we have included as rule R-1 in our domain theory in a separate section. Another rule, R-2, selects QSIM as the only solver for purely qualitative scenario models.

6 Specification of a scenario description

Another important constituent part of an organizational knowledge base is scenario descriptions. For a given enterprise we would have multiple scenario descriptions, each representing a specific viewpoint on the organization. Kaplan and Norton [14], for example, suggest a variety of perspectives which are modeled as interacting organizational units where different activities and phenomena are emphasized, depending on the particular view. A scenario description provides us with a structural configuration of the objects constituting the specific scenario being investigated, and some topological information about the organization. Usually, it also carries additional information about organizational units' behavior, such as initial states, and several simplifying assumptions that are reasonable for the given scenario. Views could include scenario descriptions in terms of cash flows and material flows, or scenario descriptions representing customer perspectives, innovation and learning perspectives, and numerous internal business perspectives, etc. The particular scenario depicted in

Proceedings of the 28th Hawaii International Conference on System Sciences (HICSS '95)
1060-3425/95 $10.00 © 1995 IEEE



Department(Finance)	Department(Marketing)
Department(Manufacturing)	Department(Accounting)
cash(NetIncome)	cash(Expense)
units(Sales)	price(Price)
price(UnitCost)	percentage(Markup)
link(Marketing, Finance, Sales)	link(Manufacturing, Finance, Expense)
link(Manufacturing, Accounting, Expense)	link(Marketing, Manufacturing, Sales)
link(Marketing, Accounting, Sales)	link(Accounting, Marketing, Price)
link(Manufacturing, UnitCost)	link(Accounting, Markup)
link(Finance, NetIncome)	

Figure 2: A Scenario Description of the CORP Corporation.

Figure 2 represents a global, highly aggregated view on the organizational subunits, and their interactions, which determine the financial performance of the organization.

Scenario descriptions provide additional information needed in the model formulation process, that is, they indicate which model fragments have to be considered when modeling a certain organizational phenomenon from a particular perspective. For example, when we model a company's net income, from the scenario description illustrated in Figure 2, the modeling system should know that model fragments for expense, sales, unit cost, etc. are all relevant pieces and should be included in the scenario model.

7 Query analysis and model composition

Unless queries are restricted in some way, they can vary tremendously in form and content. Without loosing task and domain independence, we should restrict the query language to convey only basic information. Let us suppose a user posed the query "*How does an increase in price affect net income?*" Conceptually based on natural language processing, a query elaboration procedure would analyze the issued query, and derive from it a set of ground expressions which would be passed on to the model composition module of the enterprise modeling system for evaluation. In absence of such a sophisticated query analyzer, we could simply devise a primitive query language which basically lists a number of ground expressions which permit the system to identify objects, quantities and relations of interest, where each of these has a referent in the model knowledge base. Hence, let us consider the simplified query {increase(Price), Quantity(amount-of(NetIncome))}, whose ground expressions increase(Price) and Quantity(amount-of(NetIncome)) provide the input to the model composition module.

The query indicates that we need a scenario model which computes net income. While the one ground expression Quantity(amount-of(NetIncome)) of the query does not hint to either a qualitative or a quantitative modeling approach, the other ground expression provides a clear clue for a qualitative analysis. Since the increase operator indicates a desired direction of change without further specification, it suggests a qualitative model for investigating this effect on net income in the given scenario. Because the number of modeling assumptions tends to be small, it is better to reason about combinations of modeling assumptions first, and then to select and to integrate a suitable set of model fragments. Falkenhainer and Forbus [10] advocate the application of assumption-based truth maintenance systems, and in particular the ATMS presented by [7], as a natural way to manipulate sets of modeling assumptions. The task is to find a set of modeling assumptions consistent with the preconditions of those model fragments which are going to be involved in the constitution of the scenario model.

8 Discussion and conclusion

In this paper, we have outlined a novel, comprehensive framework at a mostly conceptual level for future enterprise-wide modeling systems. In [2] we present a more detailed discussion of enterprise modeling systems. It also proposes EMS implementation strategies based on recent research in software engineering, and draws in particular on megaprogramming [23] and the concept of software agents [17]. Both approaches are designed for implementing very large software systems that contain heterogeneous subsystems. The former presents a megaprogramming language which facilitates the parallel execution of internally homogeneous megamodules by defining a supermodule, the megaprogram, which controls the interactions between subtasks. In our EMS application we define a coordinating megaprogram and a separate megamodule for each of the five functional EMS modules, as described in section 2, namely the query manager, the model manger, the candidate evaluation module, the solver, and the post processor. We also show in [2] how the concept of software agents can be applied to EMS to integrate different types of representations. Basically, it is a divide and conquer approach which delegates specific tasks to software agents which perform tasks such as analyze a query, find a consistent model, select the best from a collection of models, and solve a specific model.

We believe that future research in model building and model management for decision support in organizational environments requires more attention to related model building and model reasoning research in artificial intelligence. One purpose of this paper is to bring to bear some of the stimulating results obtained from the AI community, and to indicate how they can be incorporated into the DSS research on model building. Among the new features we have proposed, we want to highlight those two which we strongly feel map out the most promising future research directions. First, the possibility of both qualitative and quantitative model formulations, which introduces a new level of versatility to organizational model building, and which should widen the scope of computer supported decision tools considerably. Second, the application of a compositional modeling strategy to automatically build task-specific scenario models, which liberates users from having to specify special modules for controlling the modeling integration process.

We conclude by mentioning several, more specific issues which need further investigation. The success of compositional modeling depends heavily on a clear organization of the domain theory. We still need a better understanding of how to decompose an organizational environment into semi-independent components in a manner which really reflects how different parts of the organization work together to accomplish common corporate goals. More research needs to be done to develop comprehensive taxonomies and ontology which are necessary as a conceptual basis for the formulation of organizational descriptions which, truly, can be considered as interpretable knowledge units that can be used to synthesize new knowledge. Work in organizational behavior and management, such as [19] and [14], could help to structure organizational knowledge more systematically. A realistic organizational setting would take place in a distributed environment. Thus, one would like to extend our framework to a distributed computing system, where the model knowledge base is managed across functional organizational units and geographical locations.

References

[1] Addanki, S., R. Cremonini, and J.S. Penberthy, "Graphs of Models." Artificial Intelligence, 51, 145-178.

[2] Ba, S., K.R. Lang, and A.B. Whinston (1994) "Enterprise Modeling and Decision Support." working paper. Center for Information Systems Management. The University of Texas at Austin.

[3] Bhargava, H.K. and S.O. Kimbrough (1993) "Model Management: An Embedded Languages Approach.". forthcoming in Decision Support Systems.

[4] Bhargava, H.K., and R. Krishnan (1991) "Reasoning with Assumptions, Defeasibility, in Model Formulation." working paper, The H. John Heinz III

Proceedings of the 28th Hawaii International Conference on System Sciences (HICSS '95)
1060-3425/95 $10.00 © 1995 IEEE

School of Public Policy and Management, Carnegie Mellon University.

[5] Blanning, R.W. (1987) "A Relational Theory of Model Management." in Holsapple, C.W., and A.B. Whinston (eds.), Decision Support Systems: Theory and Application, Springer Verlag.

[6] Bonczek, R.H., C.W. Holsapple, and A.B. Whinston (1981) Foundations of Decision Support Systems, Academic Press.

[7] de Kleer, J.(1986) "An Assumptions-Based TMS." Artificial Intelligence, 28, 127-162.

[8] de Kleer, J., and J.S. Brown (1984) "A Qualitative Physics Based on Confluences." Artificial Intelligence, 24, 7-83.

[9] Dolk, D.R. and J.E. Kottemann (1993) "Model Integration and a Theory of Models." Decision Support Systems, 9(1), 51-63.

[10] Falkenhainer, B. and K.D. Forbus (1991) "Compositional Modeling: Finding the Right Model for the Job." Artificial Intelligence, 51, 95-144.

[11] Forbus, K.D. (1984) "Qualitative Process Theory." Artificial Intelligence, 24, 85-168.

[12] Hinkkanen, A., K.R. Lang, and A.B. Whinston (1993) "Qualitative Modeling And Decision Support: On the Representation of Organizational Systems and the Improvement of their Performance." In *Proceedings of Quardet '93: III IMACS International Workshop On: Qualitative Reasoning And Decision Technologies*, Barcelona, Spain, 16-18 June 1993.

[13] Hinkkanen, A., K.R. Lang, and A.B. Whinston (1994) "On the Usage of Qualitative Reasoning as Approach Towards Enterprise Modeling." forthcoming in Annals of Operations Research.

[14] Kaplan, R.S. and D.P. Norton (1992) "The Balanced Scorecard - Measures That Drive Performance." Harvard Business Review, 70(1), 71-79.

[15] Kiang, M.Y., A. Hinkkanen, and A.B. Whinston (1993) "An Interval Propagation Method (Rules Constraints Reasoning) for Computing Qualitatively Defined Systems." Working Paper, MSIS Department, The University of Texas at Austin.

[16] Kuipers, B. (1986) "Qualitative Simulation." Artificial Intelligence, 29, 289-338.

[17] Maes, P.(1994) "Agents that Reduce Work and Information Overload," Communications of the ACM, 35(11), 89-99.

[18] Muhanna, W.A. and R.A. Pick (1992) "Meta-Modeling Concepts and Tools for Model Management: A Systems Approach." forthcoming in Management Science.

[19] Orton, J.D. and K.E. Weick (1990) "Loosely Coupled Systems: A Reconceptualization." Academy of Management Review, 15(2), 203-23.

[20] Petrie, C. (Ed) (1992) Enterprise Integration Modeling: Proceedings of the First International Conference. MIT Press.

[21] Rickel, J. and B. Porter (1992) "Automated Modeling for Answering Prediction Questions: Exploiting Interaction Paths." In Proceedings of the Sixth International Wrokshop on Qualitative Reasoning. p. 82-95. Edingburgh, Scotland.

[22] Weld, D.S. (1992) "Reasoning About Model Accuracy." Artificial Intelligence, 56, 255-300.

[23] Wiederhold, G., P. Wegner, and S. Ceri (1992) "Toward Megaprogramming", Communication of the ACM, 35(11), 89-99.

Proceedings of the 28th Hawaii International Conference on System Sciences (HICSS '95)