A SIMULATION-BASED DECISION SUPPORT SYSTEM SBDSS FOR MANUFACTURING ENTERPRISE

Feng Shan[a] **Huang Jingping**[a] **Cen Ling**[a] & **Zhang Jilie**[b]

[a] *Department of Automatic Control Engineering, Institute of Systems Engineering, Huazhong University of Science and Technology, Wuhan, Hubei, 430074, P.R. China*

sfeng@blue.hust.edu.cn

[b] *Dong Fang Electrical Machinery Co. Ltd., Deyang, Sichuan, 618000, P.R. China*

Abstract: The SBDSS is designed to achieve a high level of performance, flexibility and adaptability, in response to meet the special needs of production and logistics management during the economic system reform era in China. It consists two subsystems, Object Library Modeller OLM and the Simulation Engine and its Manager SEM. Using SBDSS the decision makers can work out their optimal production choice under certain circumstances through scenario simulations, they can test a set of virtual organizations reflecting systems reform before a real reorganization has been taken, as well as perform a virtual manufacturing process for a new product design.

Keywords: Modeling, Simulation, Object-oriented programming, Decision support systems, Integration.

1. INTRODUCTION

In recent years, the Chinese industrial decision makers (DMs), facing with the challenges to handle the complex system convertibility problem in the dramatic social economic system reform process, require a major shift towards maturity of necessary identification of feasible reform strategies and their assessment. Tools of decision support that help DMs to learn about possible decision options and their predicated results are of urgent need. Our research efforts devoted in constructing the Simulation-Based Decision Support System SBDSS are aimed to provide such a tool. Instead of getting help from an experts team through meetings discussion face to face in the past. DMs obtain answers about the real world problems by working interactively with the SBDSS, in which an appropriately generated model or model system performs (simulates) as an experimental substitute for the modeled system itself in predetermined circumstances for observations. Although such a model can never be perfect and can not encompass all aspects of the very complicated decision situation, it is often a great help to the DM in the process of learning cause and consequences about novel aspects of the decision problem and thus gaining expertise in handling problems of a given class. This is a cost-effective approach.

In configuration of SBDSS, we were given conditions that

(1) the end-user of the system is a large state enterprise, engaged in small-batch and one-off production;

(2) there existed a hybrid computer management system, consisting many separately located subsystems, each offering a certain functional service, either for a department or for the whole enterprise. And these subsystems have been realized incrementally over many years in the past. Many subsystems are technologically obsolete, e.g. the old financial management subsystem. On the counterpart, some newly added parts are of the most up-to-date class, e.g. CAD, CAM subsystems;

(3) A cost-effective approach must be considered to keep the entire enterprise running on original management system, simultaneously to develop an integrated new system, which fully utilizes the advanced information technology, leading to the efficiency of decision making.

A simulation-based approach is chosen, because modeling and simulation are fundamental approaches to meet the high flexibility needs as given in above, and to develop applied systems of object-oriented paradigm, we have our platform and tool kit AIOOM.

2. CONCEPTUAL DESIGN

2.1 Distributed Heterogeneous Computing and Networking Platforms

As it has been described in section 1, the desired application system might involve multiple users (agents) participating in a computing environment consisting of heterogeneous and autonomous information resources. The systems that manage the resources, e.g., database management systems, might also be heterogeneous and autonomous.

These applications might be supported by distributed and heterogeneous computing and networking platforms (both hardware and software components) and have multiple administrative and access control authorities.

2.2 A Frame-work for Controlling Cooperative Agents

A software paradigm that can support such applications flexibly and reliably is a distributed cooperative task. In this paradigm, an agent supports a user, represents the user to the system, and handles complex interactions with other cooperating agents and system resources. A critical issue in such a paradigm is controlling interactions among the cooperating agents to meet the application objective, despite unpredictable user interventions and system failures.

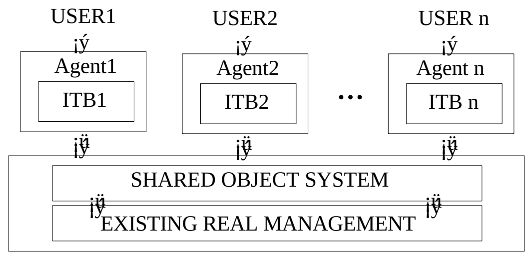


Fig. 1 Multi-agent cooperation process simulation

In Fig. 1, each distributed user being represented by a software agent equipped with an Information Interactive Transaction Block ITB. The ITBs are agent specific, and use fixed-point criterion to define a stable state of the system with respect to an agent. Thus, the Fig. 1 system supports adaptive cooperation among agents and reliable operations on shared objects stored in heterogeneous and autonomous components systems.

Group processes and decision-making can be supported as a *process and process structure* or as a *task and task structure*. Process and process-structure support include the provision of language, communication channels and means for process structuring and conducting. Tasks are traditionally supported by quantitative models and decision modeling tools, and recently also by qualitative reasoning models. The difference between these two domains can be identified by two terms: GPSS for *group process support systems* and GDSS for *group decision support systems*. The research of organizational activities, presented in this paper, concerns problems within the domain of group process and process-structure support, i.e. GroupWare, aimed to figure out an efficient way to finalize a group decision making process. And a software agent can represent each member of the group.

2.3 Computational Infrastructure and AIOOM

It has been described that SBDSS is designed to cope with various problems, which are not identifiable in detail, but heuristic solutions to it exist. We need to combine the deep knowledge of scientific principles used in simulation with the superficial what-to-do knowledge used in expert systems effectively to form a real-time learning simulation schema. We further expect that SBDSS can formulate a foundation for an integrated advisory system in CIMS environment with an autonomous goal-driven technological character.

We are confident of success, because our previous development activities has constructed the platform: Artificial Intelligence Embedded Object-oriented Methodology AIOOM, which is powerful to support the current development of SBDSS with its tools and enablers. AIOOM written in C++, naturally accommodates the heterogeneity and autonomy of large-scale, distributed systems: heterogeneity because messages sent to distributed components depend only on the component's interfaces, not on their internals; and autonomy because components can change independently and transparently, provided they maintain their interfaces. AIOOM's equipped with tools to produce knowledge-based systems, Inductive Inference Systems, Autonomous Activity Systems, Socially Organized Systems as well as their blending. AIOOM has been introduced elsewhere, refer to (Feng Shan, Li Tong *et al*. 1994).

3. SBDSS CONFIGURATION

3.1 The Application System SBDSS

The Simulation-Based Decision Support System SBDSS has been developed with our AI Embedded Object-oriented platform, AIOOM. SBDSS is designed to achieve a high level of performance, flexibility and adaptability,

It consists two separate functional subsystems: the Modeller (Objects Library) OLM and the simulation engine and its manager SEM. OLM enhances real world system abstraction by providing a natural mapping between system components being modeled and Objects of the library. Object behavior can be defined using state transition diagrams. Thus a set of Objects associated with the various linkages among them, completely change responses to in-coming messages. While the SEM is the simulation kernel, responsible for the managing the progress of simulation on one host machine. The host language of the SEM is C++, the object-oriented nature of C++ assists in realizing the correctness and software quality.

3.2 The Object Library Modellor OLM

The OLM of SBDSS provides a hierarchical basis within which a library of general-purpose simulation objects can be defined. These objects can be interconnected and specialized to construct an accurate model of any target real world system. The object-hierarchy is founded on the observation that most management processes can be viewed as composed of nodes that are interconnected in some way by communication links.

At this level a node may be viewed as a state machine. State transitions are triggered by arriving messages and this activity may produce one or more messages. The function of each node in a simulation is defined by the structure of its state machine, i.e. its nodes. Nodes are classified into three classes according to their primary function in a system, namely SOURCE, FORWARD and SINK. The three classes of nodes with their attributes are described in Table 1. We use OBFRAME to show the three classes

Table 1 The three classes of OBFRAMEs (Objects)

Slots	Name	SOURCE (i)	FORWARD (j)	SINK (k)
A:	input chs	Blank		
A:	output chs			Blank
A:	others	Arbitrary		Arbitrary
D:	datastructure			Blank
M:	Receive			
M:	Send			
M:	Get			
M:	Consume	Blank	Blank	
M:	Generate			
M:	Route			Blank
M:	Others			
D.F.	Driving			Blank
D.F.	Trace			Blank
D.F.	Stochastic	Possible		Blank
D.F.	Execution			Blank
D.F.	Others	Possible	Possible	Blank

Note: A (Attribute); D (Data); M (Message); D.F. (Driving Factor)

3.3 Simulation Engine and its Manager SEM

SEM contains a scheduler object and a route object, which manage the simulation nodes and an event list. The scheduler knows the nodes relevant to the SEM and provides the engine with interfaces to invoke the method in the node that matches the youngest outstanding event. Similarly, every future event is generated by the nodes through invoking a method in the scheduler. The router stores the interconnection scheme, which defines how the nodes of a simulation are connected together. This allows the SEM to handle the proper delivery of messages sent by one object to another.

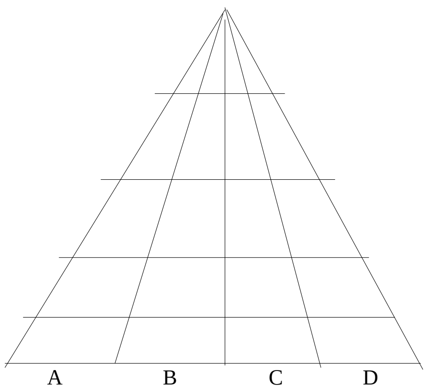
In parallel simulations, there is a set of host machines, and on each machine there is a mapping function used to forward events and messages, to the SEM containing the objects they are addressed to. At the first stage of SBDSS development, only one host machine is used.

4. APPLICATIONS

A SBDSS simulation consists of a set of nodes (OBFRAMES) and a mapping that defines how they are interconnected. Each node comprises an arbitrary number of queue objects (formed in OBFRAME, too), that represent its logical input and output channels for communication with the outside world. An execution method that is triggered by message arrival, i.e. it is “control-flow” that invoking the method. While "sending a message" to another object is a "data-flow”. To commence a simulation, a set of "active" nodes with special initialization methods will be invoked, i.e. message generator nodes actions. This abstraction allows the cooperating objects that form the model to be defined solely in terms of their functionality in the system being represented.

Our AIOOM platform has a user interface, providing a model reflecting the underlying object-services layer and a set of tools for generating interfaces to new management applications. The innovative interface permits an interactive dialogue between the user and the object system. This facilitates SBDSS equipped with a graphical simulation design tool to implement objects and linkages among them. Among others, the object services including management request brokers(for object registration and location) plus authentication and authorization services for object-access control; help to perform customizable management services implementing management domains and policies. The following Figure 2 and Figure 3 show such applications.

4.1 Case 1.



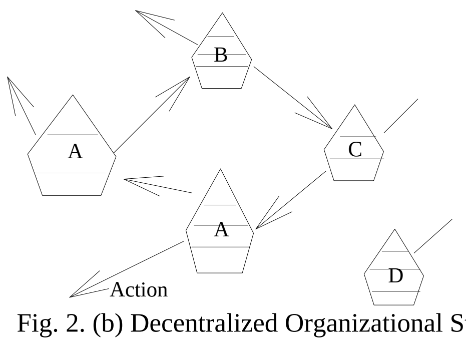


Fig. 2. (a) Monolithic System (Central Planning System) Fig. 2. (b) Decentralized Organizational Structure

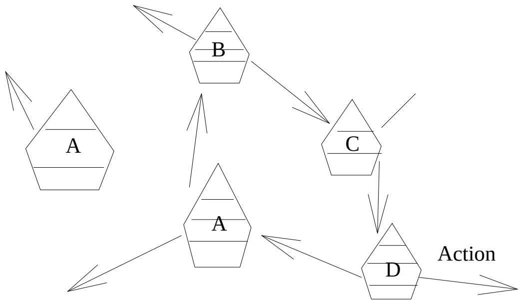


Fig. 2. (c) Decentralized Organizational Structure

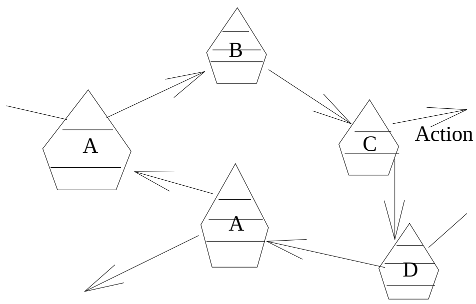


Fig. 2. (d) Decentralized Organizational Structure

Fig. 2. (a) represent the enterprise organization structure of management, it is a typical traditional management triangle. The top leader is expected to make final decision for all classes of A, B, C, D production sections, and the monitoring and control chains of the decision execution are very long.

Fig. 2. (b) is a feasible decision schema, conditional on decentralization of the decision making process. We can see only relevant production units associated with their unit leaders participate in the process. The schema is flexible and dynamic. The information flows construct a net structure to meet the rapid adaptation requirement of current market demand. Fig. 2.(c) and Fig. 2.(d) are two case specific decision procedures, and their control paths can be chosen to minimize cost through management efficiency. Using SBDSS, therefore helps DMs to choose a reasonable re-engineering schema through simulations.

4.2 Functional Scheme Generation

The SBDSS has been used to investigate the adaptable organizational structure of the objective enterprise. And Fig.2(a)-(d) are conceptual illustrations showing that decentralized organizational structure raises the efficiency of management work. Fig. 3 demonstrates the relevant functional scheme.

SBDSS supports bottom up simulation schedule from inputs suit to meet the targeted profit level, and also supports top-down schedule to figure out the quantity and quality of inputs.

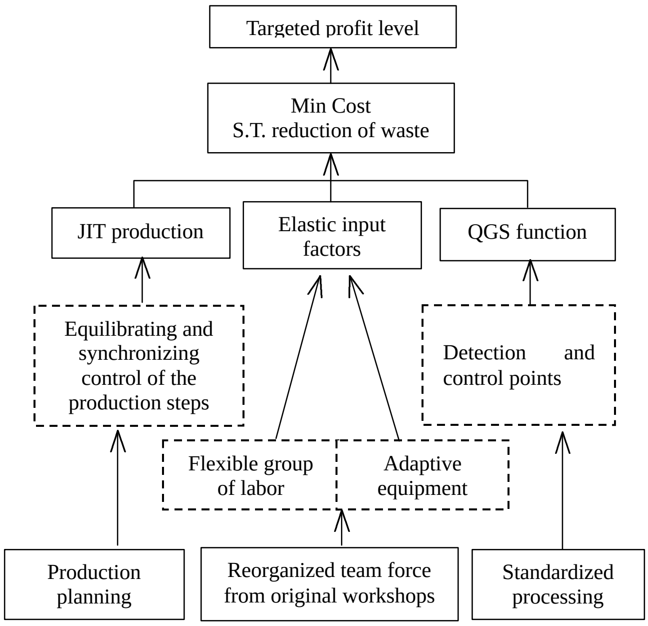


Fig. 3 Functional Scheme Generation

4.3 Case 2

From the information technological point of view, the enterprises' ends to ends management depend on handling with the increasingly complicated computer hardware and software systems, as shown in Fig. 4. The enterprise internal computer systems are those Management Information System MIS, Computer Aided Design (CAD) System, Computer Aided Manufacturing (CAM)System, Quality Guarantee System (QGS). They have communication channels with the outside world beyond enterprise for information exchange, e.g., Technological and economic information to CAD system, selling services for QGS system, etc.. Different allocation and utilization of computer resources will certainly produce a set of feasible solutions on the system computer development plan. This make it to be possible to build new functional parts at successive stages to fit both the low cost and technological advances requirement. And SBDSS can be used for development planning.

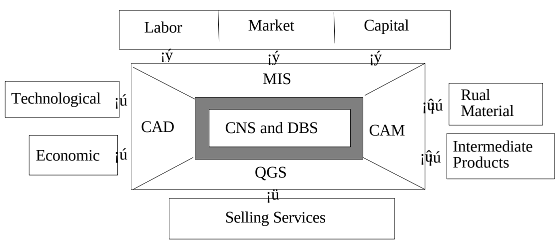


Fig. 4 Internal and Relevant Information System

5. FURTHER WORK

SBDSS has been constructed with our AI-embedded Object-Oriented platform, AIOOM.The new version of AIOOM is AIAOT in which we reformed the OBFRAMEs (objects) into AGENTs. AGENTs are autonomous OBFRAMEs. The basic key-features of an agent are autonomy, social ability, responsiveness and proactiveness. As we can see Agent-Oriented-Techniques of short AOT as a natural extension of Object-Oriented-Methodology OOM. Our new platform AIAOT provides an environment for coordination and cooperation, within which its agents can work collectively to solve specific problems. Groups of agents are identified to perform specific reasoning for a given task and decision-making responsibilities are delegated to co-ordination groups (teams) made up of these agents. Using this approach, the manufacturing system can thus be integrated through a harmonious association of individual agents and resources found at different places within the factory or in remote organizations. This conceptualization of the manufacturing enterprise involves mew paradigms for agent communities, for building the next generation of manufacturing systems. Incorporating Mediator agents to support multi-agent co-ordination can enhance collaborative behavior among the agents. Such Mediator agents have meta-level rules and higher order priorities for managing the behavior of the agents.

The architecture incorporates mechanisms for creating agent communities and co-ordinating their actions during higher-level planning (such as manufacturability evaluation, machine routing, and scheduling). During higher-level planning, the involved agents use asynchronous message communication. Following higher-level planning, the planned tasks and sub-tasks are implemented through lower-level control agents using synchronous communication and interlocking of processes among the agents. All the development work is on going.

6. CONCLUSION REMARK

(1) SBDSS provides a flexible tool for modeling complex distributed systems. With its OLM, it is capable to model real various world object systems, including computer software and hardware components.
(2) A high-level visual specification interface designed to manage the selection, editing and graphical composition of objects. This will facilitate rapid implementation of simulations.
(3) The distributed object management approach is based on current trends in object-oriented distributed systems models, application - integration environments, object-oriented databases. These trends are evolving so that the complete set of resources available on a distributed network-including computers network facilities, data and programs can be treated as a commonly accessible collection of objects. These objects combining in arbitrary ways provide new information processing capabilities.
(4) With the capabilities indicated in (3), SBDSS help to determine optimal scheme satisfying production and logistics management re-engineering requirement for system reform.
(5) The platform AIOOM has been reformed and extended into AIAOT. And based on AIAOT, next generation manufacturing systems can be modeled and simulated to support management decision making by introducing multi-agent system models into manufacturing organization, common protocols for creating intelligent agents and facilitating their inter-operation within agent communities help in providing flexibility to react to the rapidly changing manufacturing environment.

ACKNOWLEDGMENTS

Supported by the project entitled *Virtual Reality Simulation Environment Research and Application (No.96J4.3.6 JW05)* and *Dong Fang Electrical Machinery Co. Ltd.* (DFEM). DFEM is a representative large-scale state manufacturing enterprise in China. Our efforts devoted to construct the SBDSS and provide an evidence that it is a desired tool, and to make SBDSS really useful, involves deep interdisciplinary expertise for successful implementation

REFERENCES

Coomber, C.J. (1993). An Object-Oriented Procedure for Simulating Real-Time Systems In: *TOOLS Pacific'93 (Technology of Object-Oriented Languages and Systems)*, 269-278.

Dillon, T.S. ,and Tan., P.L. (1993). Object-oriented Conceptual Modeling. *Prentice Hall*

F. Manola et al. (1992). Distributed Object Management. *International J. of Intelligent and Cooperative Information Systems*, **1**, 5-42.

Feng Shan, Li Tong, et al (1994). Research Report on the Project "*Artificial Intelligent Embedded Object-oriented Methodology Development Platform AIOOM*". 1994. Huazhong Univ. of Sci. & Tech., Wuhan, China.

Feng Shan, Tian Yuan, Li Tong & Cai Jun (1996). Artificial Intelligence Embedded Object-Oriented Methodology for Model Based Decision Support, *Journal of Systems Engineering and Electronics*, **7**, 1-14.

Feng Shan, Cai Jun, Li Tong (1996), Information Technology in China In: *The Place of Information Technology in Management and Business Education* (Ben-Zion Barta Arthur Tatnall, Peter Juliff ed.) 47-52. Chapman & Hall.

Gaines, B.R. (1996). The Design of Expert Systems for Planning Flexible Manufacturing. *Proceedings Japan-USA Symposium on Flexible Automation*.

Li Tong, Zhai Fan, Li Yan and Feng Shan (1995). On Cooperative System, *Journal of Systems Engineering and Electronics*, **6**, 7-10.

Tanir, O., Sevinc, S.(1994). Defining Requirements for a Standard Simulation Environment. *IEEE Computer*, Feb., 28-34.