MULTI-AGENT MRP CLASS SYSTEM FOR PRODUCTION AND DISASSEMBLING

Krzysztof Cetnarowicz Edward Nawarecki
Grzegorz Dobrowolski

Institute of Computer Science,
AGH - University of Mining and Metallurgy
Al. Mickiewicza 30, 30-059 Krakow, Poland
e-mail: cetnar@agh.edu.pl

Abstract: Modern large companies have distributed infrastructure that brings some of new problems to be solved. Main typical tasks to be mastered are: distributed storage, transport, production planning, etc.
To solve these problems, the companies base their decision procedures on integrated management systems. Such a solution makes all strategic decisions easier and more pertinent because all required information (history, statistics, current company state etc.) is easily available.

Applications usually used by production companies are of MRP II or ERP class, and they almost always contain the closed-loop MRP class subsystem. To make production planning more efficient it is worthy to be well informed about the market for certain products and particularly about supplier’s possibilities.

Key procedures "to sell" and "to buy" (or to find the proper cooperation) remain the domain of human work.
To realize this tasks big companies create separate units, for such tasks only. High cost and low quality of realization of these very important tasks in contemporary market reality have important impact on the proper company functioning. Computer network and decentralized systems may put an alternative idea into reality.

The paper is aimed at showing design process of a multi-agent management system. Special multiagent modules are proposed to include into MRP class integrated management system.

Keywords: Decentralized MRP, Decentralized systems, Multiagent management
efficient it is worthy to be well informed about the market for certain products and particularly about supplier's possibilities.

Key procedures "to sell" and "to buy" (or to find the proper cooperation) remain the domain of human work.

To realize this tasks big companies create separate units, for such tasks only. High cost and low quality of realization of these very important tasks in contemporary market reality have important impact on the proper company functioning. Computer network and decentralized systems may put an alternative idea into reality.

The paper is aimed at showing design process of a multi-agent management system. Special multi-agent modules are proposed to include into MRP class integrated management system.

1. DISTRIBUTED MRP CLASS MANAGEMENT SYSTEMS

Modern large companies encounter a number of problems to be solved such as: distributed storage, transport, production planning, etc.

To solve these problems, the companies have based their decision procedures on integrated management systems. Such a solution makes all strategic decisions easier and more pertinent because all required information (history, statistics, current company state etc.) is easily accessible. Applications usually used by the companies are of MRP II or ERP class, but they almost always contain the closed-loop MRP class subsystem as its kernel ((Stevenson, 1990), (Greniewski, 1995)).

The subsystem comprises algorithms responsible for production management area. Communication paths in structure of the system are some kind of closed loop, which explains the name.

The modules included in the closed-loop MRP model are following:

- MRP (Material Requirement Planning) - the heart of the system, a module which computes material requirements,
- CRP (Capacity Requirements Planning) - a module which is subject to capacity requirements (both systems are initialized by the MPS module),
- MPS (Master Production Schedule) - a module responsible for production scheduling (Proud, 1994),
- SFC (Shop Floor Control) - a module responsible for management of production orders and priorities,
- PC (Purchase Control) - a module responsible for registration of purchase realization,
- IM (Inventory Management) - a module responsible for management of product and material reserves control.

All this modules based on bills of material (BOM), production paths and production demands, can generate material requirements, compute taking into consideration, stock capacity constraints and decide when to start production to realize orders, or even decide in which order production tasks should be done.

This information is very important in production planning and gives a good example of using integrated information systems in the management problem ((Stevenson, 1990), (Gaither, 1994)).

To make production planning more efficient it is worthy to be well informed about the market for certain products and about supplier's possibilities. It means that key procedures "to sell" and "to buy" (to find the proper cooperation) remain the domain of human work. High cost and low quality of realization of these very important tasks in contemporary market reality have important impact on the proper company functioning.

To collect the information, big companies created separate units, responsible for such tasks only.

Computer network and decentralized systems may put an alternative idea into reality. Special modules have been created in integrated management system which enables to automatically create network of cooperating companies and open new application for such systems.

2. MULTI AGENT SYSTEM FOR MANAGEMENT

Management of enterprises working in decentralized way, by centralized computer systems, e.g. classical systems of MRP II type leads to centralization and fossilization of organization, which results in loss of flexibility in operation ((Wooldridge et al., 1996), (Walsh and Wellman, 1998)).

Realization of management system of MRP II class as a multi-agent system makes it decentralized system with the following features:

- System developing. Adding new function to the management system is natural, needed and often applied. In agent's system this possibility is supported very efficiently. System development requires creating a new group of agents, testing agents and including them into the system. The test possibility increases the chance that the system will work correctly.
• System modification. Agent’s approach gives an easy way to make any changes in the system. The chosen modification function or modules usually demand changes be made in the body of some agent. The changes are short and simple. It releases a programmer from analyzing a whole system code. One can also easily test a new agent on the exemplary data.

• Changes in running system. Most of the modifications may be done on the running system. A new group of agents is introduced into the system and takes over the tasks from the old agents. The same possibility is in developing, that is adding new agents, which does not require stopping the system.

• Calibration. Every new element/task has its own agent, what automatically projects the size of managing business. There is a possibility of changing a number of running agents, but not a number of interactions (every agent always communicates with the same number of agents). So no changes dependent on the system size are necessary in agents’ construction. Additionally, agents as autonomous creatures may migrate in distributed environment and steadily optimize of the server load.

• System topology. Autonomous agents running in distributed environment, give the clear projection of information system topology. The company structure may be naturally divided into several units by creating separate subsystems on each of them. The tasks (agents) linked with some company unit may be computed at the local server. The feature makes managing the system more natural, enables arrangement of steady tasks in distributed system and reduce results of possible system damage.

• Cooperation with different database systems. Very important virtue of agent approach to management system is flexible cooperation with different database engines. We can obtain it by creating special access agents, specialized in serving one database or even one table. This system is completely adaptable and independent from database. Any change made in data storage structure requires only creation of an appropriate access agent (or agents).

• Modification of the MRP system to apply it to the disassembling process. The MRP system built as multiagent system may be easily put in the inverse rhythm of work to manage the disassembling process. In this case the delivered goods are disassembled into primary parts and raw materials that are to be sold for reuse in production.

3. MULTI-AGENTS MRP CLASS SYSTEM FOR DISASSEMBLING

The disassembling process may be considered as the particular production process with following characteristic properties (fig. 2):

• The Master Schedule concerns the objects for disassembling reception and remains a driving schedule for the system.

• The disassembling process is realized following the Bill Of Material (BOM) list and disassembly time chart giving as the output the parts and raw materials scheduled in the inventory records files.

• The disassembled parts are scheduled in the Gross Disassembled Parts (GDB). Then Net Disassembled Parts are obtained after realization of contracted deliveries. As the result we obtain the parts that remain in the stock and are to be sold.

• For the same kind of disassembled goods the BOM and disassembly time chart may have different form because of its technical state.

• System realizing orders is to get rid of disassembled goods and deliver it to clients.

The disassembling process is very difficult and may take a number of different ways of realization. So, because of that the system used to supervise it must be very flexible and easy for modifications and extensions.

The very various state and kind of disassembles parts makes problem with delivery of it. For this purpose the multiagent network system for delivery contracting and realization may be a crucial process for the whole disassembling-process success.
Although functioning of a given enterprise is driven by a computer system (e.g. of MRP II class), even including also some supplies from fixed co-operators, the process of starting cooperation, searching for suppliers and what is more important - for recipients, is performed out of the system by people. It is an expensive, slow and not reliable process, which is liable to various disturbances (delays of supplies, corruption when concluding contracts, etc.)

At the same time, the system controlling work of an enterprise, functions on a computer which may be (or even is) connected to computer network (MAN, WAN) with appropriate security measures be kept. It enables exchange of information with other, similar systems. The possibilities may be used for starting and realizing cooperation with other enterprises and automatic creation of groups of cooperating enterprises which are subject to evolution according to their needs.

Searching cooperators in network, realization of cooperation and dynamic modification of cooperating groups may be successfully realized by multiagent systems (fig. 1).

In algorithms of agents’ functioning, one may use existing, already applied algorithms and procedures such as ((Cetnarowicz et al., 1997), (Cetnarowicz, 1996)): classic streamlining algorithms, procedures based on the "order-planning" model, or more advanced SCH (Supply Chain) algorithms were used to realize the cooperation very close (usually when companies have some capital relations), more advanced algorithms based on agents cooperation, agents negotiations and multiagent teams.

First of all, potentially cooperating systems have to allow other systems to send them an invitation for cooperation. In the multiagent systems, a broker-agent (Ba) fig. 3, a special agent who migrates through the network, realizes the invitation.

When the broker-agent finds the company that is interested in cooperation in certain industry, it starts to develop cooperation process. At the beginning a technical specialization of the produced/ordered commodity has to be exchanged. If the specialization is accepted for both sides, the second step may be started, namely negotiation of the cooperation conditions. In most cases, at this level human interaction is necessary.

Anyway, we may find many other cases (directly determined roles at the beginning of the cooperation) that an autonomous agent can serve perfectly on its own. When the roles are acceptable for both cooperating partners, the real cooperation may start.

The agent comes back to the parent system and reports cooperation conditions. To acknowledge the cooperation, also the second system sends the ACK-agent, which asks the first one whether the cooperation conditions are accepted. After this, both companies set the new cooperation into the cooperation-list. The list will be searched each time when the cooperation is planned.

After the cooperation has been set, the client system may generate an order. The IM agent usually initiates the ordering process, but at first the client system has to choose the most appropriate producer system from the cooperation-list. The decision may be based on many circumstances such as: the lowest price, the highest quality, the most promising cooperation perspective, the longest credit period, or any other parameters. All the information is placed into special cooperation-table, which is refreshed automatically after any cooperation process.
The table contains all the information about the cooperation system (industry, capabilities, prices, priority, cooperation history). The information is received from communication agents (e.g. broker-agent, order-agent, ACK-agent, ERROR-agent, END-agent) or is set by a human-operator (fig. 4).

When the producer is chosen, the order may be generated. An order contains such information as: technical specification of the product, required quality, product quantity, delivery date, price (taken from the cooperation-table), others (if necessary).

The information is placed into an order-agent, which goes to a destination system (producer). When the producer receives an order and the data is acceptable, the producer system sends back the ACK-agent to the client system.

In case the data is unacceptable, the system sends the ERROR-agent, which contains information about the incorrect data. In this situation the client simply refreshes the cooperation-table and starts the producer selection process from the very beginning.

If the client system receives the ACK-agent, it notes that an order will be fulfilled. It should be remembered that in the ordering process many other producer units have to be used. One has to use the MPS to plan a new production package. Also the MRP and IM modules have to check if the material requirements are practicable. In certain cases the system checks in the CRP module if the order may be realized at the current capacity.

During the production cycle that goes right, no information is exchanged. However, in case of any trouble, the producer system sends the ERROR-agent to the client. The agent is given information about the problem, which has occurred. From this moment, the client may accept the new conditions or quit the order and find a new producer. In more complicated situations, additional negotiation may be necessary. The negotiations are very similar to the cooperation development process. Then, the new cooperation conditions are updated in the cooperation-table.

When an order has been realized, the producer system sends an agent with the notification and waits for the reply. The client system, which has received the notification, sends the ACK-agent to acknowledge that it has noted the product is ready. From this moment other units (IM in most cases) take control over the product.

When cooperation becomes more intensive, cooperator roles may change. Companies have a lot of experiences in mutual cooperation and have good knowledge of each other.

The experiences increase trust between partners. Having confidence, one of the companies may suggest some changes in cooperation conditions. The most often changes are in the following: an amount of the maximal credit, special prices, a higher delivery priority.

In the multiagent management system the next special agent implements the process. Intensification of cooperation may be done in two ways:

A client sends the agent periodically, from the time the cooperation is started. The agent asks the producer-supplier system if cooperation conditions may be changed. When the answer is positive, the producer sends the renegotiations-agent, which contains new cooperation conditions.

The renegotiations-agent is sent to the client, when a producer decides that the cooperation roles may be changed (fig. 5). The agent notifies a client about new cooperation conditions. Such a case is more natural and does not generate unnecessary network load. Whatever way is chosen, when a producer agrees that the conditions might be changed, a renegotiations-agent is generated.

The process may influence a variety of cooperation conditions. The most important fact is that anything a producer suggests, a client cannot refuse. The client simply refreshes his/her own coopera-
Fig. 6. The cooperation ending (Ea - Ending agent)

If new conditions are not satisfactory, a client may choose the other producer. The client system sends the ACK-agent to notify that it has received the new conditions.

There are many cases when the cooperation goes wrong, namely payment delays, delivery delays, too low product quality, or many other abnormalities. When problems are serious enough, one of the cooperating sides may decide to break the relation. The system, which wants to quit the cooperation, sends the END-agent (fig. 6).

The agent informs the other system about the reasons why the cooperation cannot be continued. The second system also sends the END-agent to acknowledge the end of the cooperation. The system that receives the END-agent automatically notes in the cooperation-table that a certain producer-client is blocked and cannot be used in the future. In some cases, blocked positions in the cooperation-table may be unblocked. There may be a few reasons for this: too small number of other producers-clients, serious changes in blocked company conditions or decision of a human operator. Only a human operator may execute this operation. When the current cooperation company is unblocked, the system sends the broker-agent and starts negotiation of cooperation conditions from the very beginning.

5. CONCLUSION

A multiagent virtual enterprise presented in the paper enables automatic self-organizing cooperation via network.

Cooperation between different systems is possible owing to negotiations. Management system based on multiagent technology gives possibility of supporting decision processes in enterprises.

Such a system has numerous advantages. Among them it is worthy to mention the following ones:

- lower probability of corruption,
- quicker reaction to changes of cooperation conditions,
- better efficiency of functioning (time and cost),
- flexibility,
- easiness to introduce modifications,
- ability of making modifications on running system.

The approach opens new possibilities for production management and particular to join the disassembling process to the large market.

The flexible, optimal negotiations realized via network with the use of the decentralized - multiagent systems could make the disassembling process profitable.

6. REFERENCES


