# Building Agent-Based Systems in a Discrete-Event Simulation Environment

Botond Kádár[1], András Pfeiffer[1,2], and László Monostori[1,2]

[1] Computer and Automation Institute, Hungarian Academy of Sciences, H-1111 Budapest Kende u. 13–17, Hungary
[2] Department of Production Informatics, Management and Control, Budapest University of Technology and Economics, H-1111 Budapest Müegyetem rkp. 3-9
{kadar, pfeiffer, monostor}@sztaki.hu

**Abstract.** The paper outlines a discrete-event simulation environment for modeling agent-based manufacturing systems. Exploiting the advantages of a general discrete-event simulation package, in the developed system agent-based features are directly included in the simulation environment providing the possibility to build agent-based models inside the simulation. The paper describes the agent-based functionalities of the system by presenting the communication mechanisms and predefined collaboration protocols. The modeling system implements the heterarchical control concept that is based on the contract net protocol.

## 1 Introduction

Research in *multi-agent systems (MAS)* considers the behavior of a collection of autonomous nodes aiming at solving a given problem. A MAS is defined as a loosely coupled network of problem solvers that work together to solve problems that are beyond their individual capabilities. The nodes of the system are called *agents* which are self-directed software objects with their own value system and capability to communicate with other agents. Advantages of multi-agent systems include: self-configuration, scalability, fault tolerance, emergent behavior, and massive parallelism [1]. Multi-agent systems are used for heterarchical control, where the complete decision process is performed without any form of hierarchy. A number of researchers attempted to apply agent technology - among other fields - in different segments of production systems. Examples exist for enterprise integration, supply chain management, manufacturing planning, scheduling and control, materials handling and holonic manufacturing [2].

In most of the cases, e.g. agent-based manufacturing control or material handling and manufacturing logistics the new approaches were tested in agent-based simulation environments where the simulation and the agent-based part of the system were separated. The simulation substituted the real manufacturing system and the heart of the agent-based mechanism was developed in a separate system [3], [4]. Today several different agent building frameworks, middleware are available providing basic functionalities for building distributed applications.

M. Pěchouček, P. Petta, and L.Z. Varga (Eds.): CEEMAS 2005, LNAI 3690, pp. 595–599, 2005.
© Springer-Verlag Berlin Heidelberg 2005

However, none of these frameworks gives functionalities of objects suitable for modeling manufacturing systems. The paper outlines a discrete-event simulation environment which enables the modeling of agent-based manufacturing systems. Exploiting the advantages of a general discrete-event simulation package, in the developed system agent-based features are directly included in the simulation environment providing the possibility to build agent-based models inside the simulation.

## 2 The Agent-Based Manufacturing Simulation System

The proposed model is based on a general object-oriented simulation system consisting of material, personnel and information sub-systems. This environment provides an object library that contains a number of basic objects. Combining and customizing these basic elements user-defined objects can be created that enable the construction of application-specific libraries.

The agent-based manufacturing model applies a fully heterarchical architecture including two type of agents. Based on the physical decomposition approach resources are the basic active units of the system that can perform different tasks. They are represented by *Resource agents*. An *Order agent* (the other basic unit of the system), on the other hand, includes functions such as order management and task dispatching. The parts passing through the system are active information elements holding all the relevant information about themselves they assist the agents in their actions.

### 2.1 Communication Between Agents

The whole communication process between agents is modeled inside the simulation and the agents exchange only string based messages. For this purpose a specific communication protocol was developed. The protocol specifies the form of a message and the possible exchangeable amount of information. The template of a messages is as follows:

msg_ID|delivery_time|ack|sender|type|name|param1|param2|
param3|param4|param5|param6|receiver

- msg_ID: unique identification of the message,
- delivery_time: the point time when the message was sent,
- ack: flag indicating whether the sender needs acknowledgement of the reception,
- sender: the name of the sender agent,
- type: the type of the message, e.g. TASK_ANNOUNCEMENT, BID, etc.
- name: optional and type specific information for the identification of different messages belonging to the same type,
- param$n$: optional and type specific information, transporting the basic knowledge between agents,
- receiver: the name of the receiver agent.

The action flow in the discrete event simulation corresponds to discrete actions in the model that are mainly bounded with real physical parts. This means that the simulation controller continuously updates a list of points in time when part creation, part movements, starting of a process on a part, part deletion, method calls, etc. are occurred. A minor drawback of such a system is that changes in information blocks are not treated as discrete events. The message exchange between two agents is a typical information change that does not enter events automatically in the event list of the simulation controller. We overcame this drawback by applying *"active" message boxes* that are responsible for the management of the sending and the receiving processes. In this construction each message box sends and receives messages in predefined intervals. The intervals for incoming and outgoing messages are treated separately and can be adjusted by for each individual agent independently.

## 3 Resource Allocation by Using the Contract Net Protocol

The resource allocation process in the system is based on the well-known contract net protocol. Briefly, the order agent announces the tasks to be performed, the resource agents compete for the announced tasks by bidding and the order agent selects the resource to perform the task (Fig. 1).

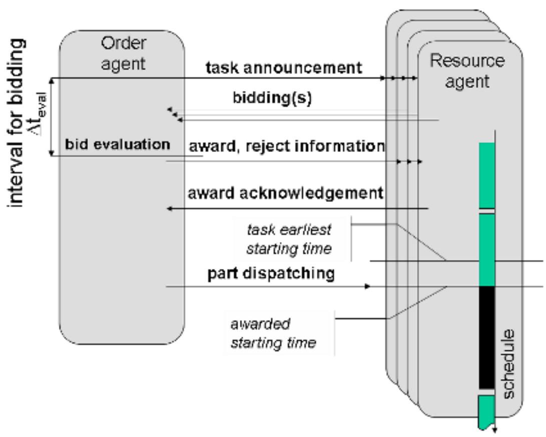


**Fig. 1.** Task announcement, bidding, awarding and dispatching cycle

Each task announcement message includes a bid evaluation interval (teval) after which the bids are not accepted (Fig. 1). During the determination of this interval the order agent has to take the communication replay delay of each agent into account. To have a smooth flow of task execution it is essential to adjust the communication parameters of the agents by setting the reading and sending intervals to be considerably smaller then the average task processing time.

Having received task announcements the resource agents construct and submit bids according to their local state variables. The resource checks the requirements of the operation and creates a bid only if it is capable to perform the operation. Each resource has an inner parameter indicating the bidding performance. This is a state variable of the resource through which the bidding behavior of the agent can be manipulated. The bid will include this earliest starting time, the processing time of the operation and the calculated cost.

The objective in the bid evaluation procedure can be the minimization of production costs, minimization of job tardiness, minimization of makespan or weighted combination of the above or similar factors. The weights of the objective functions can be dynamically adjusted on the basis of the system state and external conditions. Different weights and different rules will result in different control strategies and system performances. These can be regarded as parameters of the manufacturing system and they can be inspected from outside providing an exercising environment. All the bids submitted in the bidding period are evaluated and sorted according to the objectives. The task is offered for the resource that sent the most advantageous bid, while the others are informed about the rejection. The negotiation is completed when the awarded resource accepts and acknowledges the receiving of the announced task.

### 3.1 The Adaptive Behavior of Agents

In this concept, a purely local adaptation scheme was developed and included in the system. Considering the resource objectives, the resource agents can adjust their cost factors according to their local state variables and previous observations. Each agent incorporates a rule base with which it can locally decide on the cost factor to be applied for an announced task. The preconditions of these rules are the utilization of the resource and the ratio between the won and lost bids. The data about the bidding history are stored locally for each agent in the table of machine abilities and history. Simulation runs were performed to find and adjust the right thresholds in the rules for different manufacturing systems and loads. A detailed description of the developed adaptive algorithm and simulation results are presented in [5].

## 4 Conclusion

The paper presents a pioneer work to integrate agent-based mechanisms in a general discrete-event simulation environment by building templates that provide the well-known agent features directly in the simulation environment. The main aim of the work was to build a general and flexible modeling environment that enables the easy creation and evaluation of different agent-based manufacturing systems. Further work was initiated in order to integrate other collaboration templates in the system.

## Acknowledgement

The research was supported, partially, by the projects "Real-time, cooperative enterprises" in the frame of the National Research and Development Programme by the Ministry of Education, Hungary (Grant Nos. 2/010/2004). A part of the work was covered by the National Research Fund, Hungary, Grant Nos. T043547, T046509 and T049486. Botond Kádár greatly acknowledges the support of Bolyai János Scholarship of the Hungarian Academy of Sciences significantly easing the contribution.

## References

1. Baker, A. D.: A Survey of Factory Control Algorithms That Can Be Implemented in a Multi-Agent Heterarchy: Dispatching, Scheduling, and Pull. Journal of Manufacturing Systems, **17**/4, (1998) 297–320
2. Shen, W., Norrie, D.H.: Agent Based Systems for Intelligent Manufacturing: A state-of-the-art survey Knowledge and Information Systems, an International Journal, **1**/2, (1999), 129–156
3. Brennan, R. W., William, O.: A simulation test-bed to evaluatemulti-agent control of manufacturing systems. Proceedings of the 2000 Winter Simulation Conference, (2000) 1747–1756
4. Pěchouček, M., Vokřínek, J., Bečvář: ExPlanTech: Multiagent Support for Manufacturing Decision Making. IEEE Intelligent Systems **20**/1, (2005) 67–74
5. Kádár, B.; Monostori, L.; Csáji, B.: Adaptive approaches to increase the performance of production control systems, Proc. of the 36th CIRP Int. Seminar on Manufacturing Systems, June 3–5, 2003, Saarbrcken, Germany, pp. 305-312 and CIRP Journal of Manufacturing Systems, **34**/1, (2005) (in print).