HIDES: Towards an Agent-Based Simulator

Jens Henoch and Heinz Ulrich
Institute for Operations Research
Swiss Federal Institute of Technology, Zürich
Clausiusstrasse 47, 8092 Zürich, Switzerland
{henoch, ulrich}@ifor.math.ethz.ch

KEYWORDS
Agent-Based Simulation, Agent-Based Management-systems in Logistics

ABSTRACT

The simulation platform HIDES is particularly designed for the simulation of logistics systems. The most important characteristics of this discrete-event simulator are the reference model for logistics systems providing a small number of simple generic elements as modeling basis and the strict separation of physical and logical level in the modeling process. The planning and control task in logistics is of increasing complexity, hence distributed solution approaches, as the multi-agent concept, offer promising perspectives. For developing purposes a test-bed for agent-based simulation is conceived by adding a third level to the simulator kernel, the management level. This management level comprises a management system including an agent community, where agents are in charge of the operational tasks in the logistics system.

INTRODUCTION

Planning and control of logistics systems in today's economic environment is a demanding task due to the system characteristics we have to cope with. The system has to react and adapt to an increasing local extension and a dynamic fast changing environment. The new development in information technology provides powerful facilities for communication and information processing and offers promising perspectives for supporting the management task effectively. Static decision rules alone are no longer satisfactory for the planning and control process and a multi-agent approach is therefore indicated. For the evaluation of management concepts using multi-agent support we intend to provide a simulation platform enabling the testing of different organizational structures and management strategies as to their performance and efficiency.

This simulation platform is based on our simulation kernel HIDES, a discrete event simulator for logistics systems with the extension of a management system. The management system is added to the logical and physical levels of HIDES and allows introducing a multi-agent system.

The paper is organized as follows. First, we present the logical architecture of the simulator. In the subsequent section we introduce two generic meta types of agents. Then we describe the software concept for the realization of the simulator and finally, we give some concluding remarks and an outlook.

ARCHITECTURE OF THE AGENT-BASED SIMULATOR

In logistics, management system design is a process which proceeds from concept to realization in several steps with frequent iterations. Before final implementation, dynamic aspects of the system can only be tested in a model using simulation in an experimental environment. Simulation is therefore an important method applicable in various contexts.

The development of simulation models is a very time-consuming task, especially since the necessary frequent adaptation of the models within a dynamic development process demands a well-designed flexible model base to perform this task efficiently. Therefore, we define an architecture for a simulation platform, which is able to substantially support the design process for management systems in logistics.

Reference Model

The architecture of a simulation platform in a specific application field should be based on a meta model of this environment, called "reference models" as Schmidt (1996) proposes. As reference model for logistics systems we assume the following abstraction:

We define three levels, *a physical, a logical and a management level* as illustrated in figure 1. This distinction allows a separation of the physical part of the system from its planning and control part. As we focus in logistics management on planning and control, the physical level often remains unchanged whereas model modifications are necessary only on the other levels. By this separation, the model design process is much more transparent.

On the physical level, the static structure of the system and the dynamic sequence of the physical events is represented. The relationships are viewed as a directed graph with two types of nodes, stations and queues. Other important model elements are resources introduced as abstract elements. They can be allocated dynamically and dispose information about their availability and their behavior. Dynamic elements moving in the graph, are introduced as entities which are characterized and identified individually.

The logical level comprises the definition of the processes to be executed on the physical level, the necessary information flow to fulfill this purpose and their logical control. The nodes on this level are logic elements each in charge of triggering one specific station. They control the process flow in a decentralized way. The entities arriving at the associated station are handled by the logic element according to a general default proceeding or an individually defined protocol containing activities such as to generate, duplicate, split, modify and remove the arriving entities. In addition any algorithmically definable control logic can be implemented. The information is stored partly globally and partly locally in stations and entities. Whereas the global information is globally available, the locally stored information is only available in the local environment.

The extension of the simulator HIDES from discrete event-simulation to an agent-based simulation is realized by introducing a management level. This additional level provides the basis for developing the coordination process among the agents, which formally regulates competence and responsibility of the individual agents. Agents may interfere on the logical level in the triggering process of a station or directly to the information base of moving entities or resources.

Management System

The logistics system is embedded in a dynamic environment developing in short and long time evolution. A permanent interaction between environment and logistics system takes place. Our approach for an appropriate handling of the management task is based on the cybernetical management theories for socio-technical systems. We refer as theoretical foundation for a cybernetic structure to the Viable System Model (VSM) by Beer (1981) and the Model of Systemic Control (MSC) developed at the University of St. Gallen (Schwaninger 1994).

Beer pointed out, that a system is only viable if it has a specific management structure (Invariance Theorem). According to the proposed Viable System Model a set of management tasks is distributed to 5 systems to ensure the viability of any social system:

System V is the top normative decision level (policy-making) of the whole (logistics) system.

System IV is the two-way link between the primary activities and its external environment.

System III is the controlling unit of System I–III. It has to assure the internal stability and to optimize the allocation of resources.

System II coordinates and regulates the partly autonomous sub-systems.

System I controls and optimize the performance of the partly autonomous sub-systems on the short-term basis.

A logistic system has to evolve in the long run and to meet short-time related goals as well. Different control variables have to be considered in order to pursue these different goals. According to the Model of Systemic Control three logical management levels are introduced,

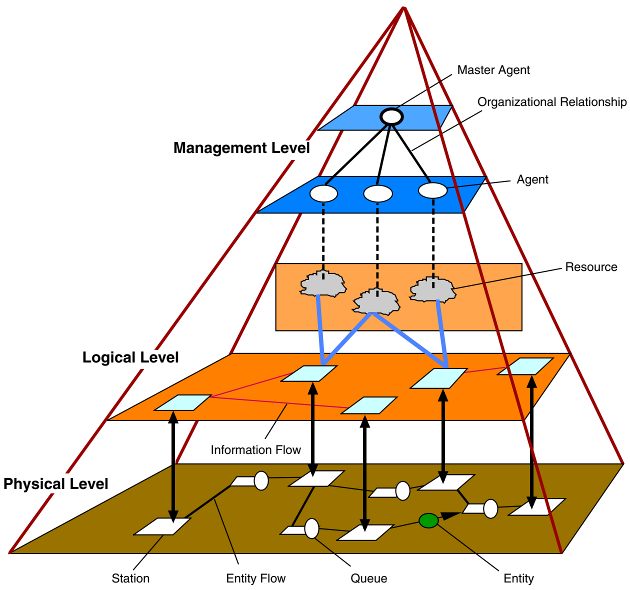

Figure 1: Logical System Architecture

where system I-III of the VSM forms the operational, IV the strategic and V the normative management level.

The levels are tied together through a feed-back control cycle, where control variables and parameters of the higher logical level exert a pre-control function in relation to the lower ones. Moreover, actors and agents[1] on higher management levels have to be able to absorb residual variety, i.e. those issues, which can not be brought under control at lower management levels (cf. Schwaninger (1998)).

For the purposes of an agent-based management, we concentrate on operational aspects of management only. Therefore tasks on higher management levels have to be accomplished by actors. System III of the Viable System Model is therefore an interface between the multi-agent system and human operators.

The tasks on the higher management levels consist of on the one hand, of the external coordination of the system with its environment on a long-term basis and on the other hand, of internal coordination of the actors of the lower level in their effort to reach their operational goals. In doing this, the effort to meet the global system goals should always be the guideline.

The operational management level comprises the management skill to solve specific tasks in the running process in view of the given operational goals and the reaction to events in the environment on a short-term basis. The tasks on this level are now allocated to the multi-agent system.

The management system hosts the agent community with their internal organization and is interfering in the logical level whenever management tasks arise. Without interference of the management system the simulation runs in a predefined mode as before.

[1]We refer to actor as an individual and an agent as a software entity.

Agents in Logistics

Three types of agents in the domain of logistics can be identified (similar to Van Brussel et al. (1998)), which are described as followed:

Resource Agent A resource can be either allocated by applying simple rules or managed in a more sophisticated manner by resource agents. Resource agents can be associated to groups. As long as there are more than one agents in a group, a master agent can be created. The main task of this agent is to coordinate the activities of its group.

Product Agents A product agent holds all process information, which was formerly kept in the attribute list of an entity. It knows the components needed to be able to complete a task. Furthermore, it has to assure that a task is correctly executed and in a sufficient way.

Order Agents An entity can be regarded as an order agent. It holds all relevant customer specific data, such as due dates. It keeps track of the logistic task and makes sure it is performed within the time frame. Similarly to resource agents, there is a possibility to pool orders in a group with a responsible group master. If the logistics system isn't able to accomplish all orders of a group within the due date, the group master has to assess the situation and to take an appropriate decision.

AGENTS IN HIDES

Franklin and Graesser (1996) reviewed various definitions of the term agent. The most general way to describe the notion is derived as stated below:

Agents are software entities, which are dedicated to a specific purpose and carry out some set of operations in order to accomplish tasks.

From a logistics management perspective this definition is too simple. Agents taking responsibility in fulfilling the operational goals as well as agents supporting these tasks in a computational way are needed. These thoughts motivate the introduction of two generic meta types of agents in the logistic domain. *Management agents* pursue goals with respect to their environment and their defined action space, whereas their contractors, the *service agents* solve well specified tasks autonomously. Management agents need various algorithmical problem solving methods, such as scheduling of tasks with respect to certain conditions, so it makes sense to delegate these kind of tasks to computational agents particularly designed for such purposes. With this task distribution, management agents are able to focus on decision related problems.

Management Agents

Management agents are the central part of a management system and defined[2] as follows:

Management agents in logistics are software entities for meeting operational goals on behalf of a human actor or another managerial agent with some degree of independence or autonomy, and in so doing employ some knowledge or representation of the user's goals or desires.

More explicitly: management agents are goal directed, pro-active, take goal responsibility, make decision, and have a model of their environment. They act autonomously, but the actions are constrained by the provided information and models[3](e.g. from a supervising agent). A management agent needs, therefore, the following information:

- a goal to pursue
- its skills and behavior
- model of its environment
- its role in the agent community (e.g. supervising agent, its sub-agents, collaborating agents)
- communication and cooperation protocols

Since we are in a logistic domain, where the coordination of joint actions play an important role, a management agent must exhibit the following properties:

- act in a collaborating manner
- apply various problem solving strategies
- communicational abilities
- methods for solving conflicts among its sub-agents
- capabilities for goal and model building, which are used by its sub-agents

The previously described resource, product and order agents (see section Agents in Logistics) have to show all of the above mentioned attributes, therefore they are management agents.

Service Agents

As mentioned earlier in this paper instances of this agent type mainly work on behalf of managerial agents:

Service agents are contractors of other agents with special computational capabilities without having any representation of their client's goals.

Typical tasks for service agents in the domain of logistics are:

- computing schedules
- monitoring the logistic system for special events, e.g. machine breakdown
- computing performance data related to logistics such as machine utilization

Moreover, service agents are free to split tasks among sub-contractors.

[2]This definition was adapted from the IBM definition for intelligent agents (see Franklin and Graesser (1996)).

[3]Gasser (1992) outlined the benefits of using models in DAI Systems.

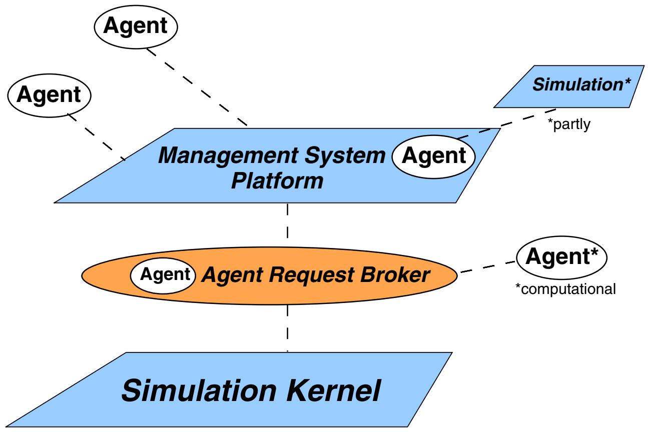

Figure 2: Software Architecture

SOFTWARE ARCHITECTURE

As programming environment, VisualWorks[4] was chosen, which is based on the object-oriented programming language Smalltalk-80. Smalltalk supports very well the software development of agents, because agents are quite similar to objects (see Shoham (1993); Burmeister (1996) for further discussions on this topic).

As we claim to have a clear separation between the general modeling (physical mapping) of a process and the control unit of that process, the simulation and the management system run distributed in separate images[5]. To connect these two parts an additional service platform, the *Agent Request Broker*, is introduced. The intercommunication between the different images is provided by Opentalk[6]. Furthermore, computational demanding processes can also run distributed in other images as indicated by the dashed lines in figure 2.

The platform consists of three major components (depicted in figure 2) provided as parcels[7]:

- Simulation Kernel
- Agent Request Broker (ARB)
- Management System Platform (MSP)

In the subsequent subsections the mentioned packages are described in more detail.

Simulation Kernel

The simulation kernel (cf. Graber et al. (1993)) allows the modeling of the physical and logical level of logistic systems and the simulating of the corresponding model. The implemented classes are partitioned into two parcels. HIDES-Core contains all classes required to run the simulation, whereas the classes in HIDES-Interface enable the user to develop the simulation model as well as display graphical simulation results. The partitioning facilitates the code maintenance and improves the simulation performance by reducing the amount of objects in the case of simulation triggered by agents. Management agents can use their own simulation models for evaluating future consequences of their decisions, which can improve planning and control (Lüthi et al. 1996).

Management System Platform

The management agents are residing on this platform. A generic framework for implementing management systems will be provided in the future. It should contain the elements listed below:

- abstract management agent models
- organizational models
- organizational learning models
- cooperational modules
- variables to measure the organizational fitness
- feed-back control cycles

Agent Request Broker Architecture

The Agent Request Broker is an interface between the simulation kernel and the management system platform. Basically, it is a service provider, independent from the actual simulation model and the implemented management system.

The following main services of an ARB provided for managerial agents can be identified:

- mediate service agents for problem solving
- storing of shared information
- running additional simulation models to support decision making
- message protocols for inter-agent communication
- intervene in the simulation

An essential part of the Agent Request Broker Architecture is a blackboard (Craig 1995; Engelmore and Morgan 1988), a hierarchically organized database with knowledge sources, around which all problem solving activities are centered.

The main reasons for choosing a blackboard architecture for realizing agent communication and problem solving are:

- The use of a blackboard architecture is a vehicle for integrating multiple sources of knowledge to solve complex problems.
- The blackboard enables an efficient information exchange for agents. Access and write restrictions for data can be introduced to realize a selective availability of information (e.g. coalitions of agents).
- The blackboard receives all messages either from the process or from the external environment. We've implemented two messaging types: broadcasting and peer-to-peer communication. The usage depends on the access status of a message.

[4] VisualWorks® is a product of Cincom Systems, Inc.

[5] An Image contains the compiled form of each method, as well as the initial bitmap that appears on the screen when the system is installed, and all the other system objects.

[6] Opentalk is a distributed programming environment for VisualWorks and provides a pure Smalltalk messaging protocol and is based on the CORBA reference model.

[7] Parcels are packages of Smalltalk objects that work together as functional units and can be stored as external files.

- Management agents can post computational problems to the blackboard in order to get them solved by the service agents.

SUMMARY & OUTLOOK

The existing simulation kernel has been extended with a management level. The resulting tool enables the evaluation of different management concepts and strategies with two key advantages:

- The model design process is much more transparent, due to the separation of the physical and controlling level.
- The performance analysis of management systems is almost independent from algorithms and heuristics, because the service agents can be contracted from all management agents.

After completion of the simulation platform, we will work on practical applications in the field of shop-floor control in microchip manufacturing and on mail distribution by a group of robots in a building. In the future we'd like to extend HIDES as a decision support tool for humans.

Further information about the ongoing research project are provided on the Web: http://www.ifor.math.ethz.ch/projects/HIDES.html.

REFERENCES

Beer, S. 1981. *Brain of the firm*. Cichester: John Wiley & Sons.

Burmeister, B. 1996. Models and Methodology for Agent-Oriented Analysis and Design. In Fischer, K., ed., *Working Notes of the KI'96 Workshop on Agent-Oriented Programming and Distributed Systems*, number D-96-06. DFKI Saarbücken.

Craig, I. 1995. *Blackboard Systems*. Ablex Publishing Corporation.

Engelmore, R., and Morgan, T. 1988. *Blackboard Systems*. Addison-Wesley Publishing Corporation.

Franklin, S., and Graesser, A. 1996. Is it an agent, or just a program?: A taxonomy for autonomous agents. In *Proceedings of Third International Workshop on Agent Theories, Architectures, and Languages*. Springer.

Gasser, L. 1992. An Overview of DAI. In Avouris, N., and Gasser, L., eds., *Distributed Artifical Intelligence: Theory and Praxis*. Kluwer Academic Publishers. chapter 1, 9–30.

Graber, A.; Ulrich, H.; Schweizer, D.; and Zimmermann, A. 1993. A Highly Interactive Discrete Event Simulator designed for Systems in Logistics. In Verbraeck, A., and Kerckhoffs, E., eds., *European Simulation Symposium Proceedings*.

Lüthi, H.-J.; Ulrich, H.; and Dürig, W. 1996. Innovation by simulation using the example of an automated workcell. *Central European Journal for Operations Research and Economics* 4(2-3):135–154.

Schmidt, B. 1996. Referenzmodelle. Simulation in Passau, Heft 2, pages 5-7.

Schwaninger, M. 1994. *Management-Systeme*. Campus.

Schwaninger, M. 1998. Managing Complexity - The Path Towards Intelligent Organizations. Contribution to: Complex System Revisited.

Shoham, Y. 1993. Agent-oriented programming. *Artifical Intelligence* 60(1):51–92.

Van Brussel, H.; Wyns, J.; Valckenaers, P.; Bongarts, L.; and Peeters, P. 1998. Reference architecture for holonic manufacturing systems: PROSA. *Computers In Industry, Special Issue on Intelligent Manufacturing Systems* 37(3):225–276.

BIOGRAPHY

Jens Henoch studied Chemical Engineering at the Swiss Federal Institute of Technology Zurich (ETH). He joined the Institute for Operations Research (IFOR) in 1997. His research interests center around modeling of agent-based management systems in logistics and simulation in general. He has experiences in simulation of chemical, logistic and forest harvesting processes.

Heinz Ulrich is a senior research associate at the Institute for Operations Research of the ETH. After a diploma in mathematics and supplementary studies in business administration he received his Ph.D. at the ETH. His primary interest is the transfer of theoretical knowledge into practice. He has successfully realized projects of simulation and decision support systems in the field of production planning and control and is now involved in several research projects in CIM-environments.