A Discrete Event Simulation Framework for Agent-Based Modelling of Logistic Systems

Bernd Page[a], Nicolas Knaak[a], Sven Kruse[a]
[a]University of Hamburg
Faculty of Informatics
Vogt-Koelln-Str. 30
D-22527 Hamburg
page@informatik.uni-hamburg.de
knaak@informatik.uni-hamburg.de
kruse@informatik.uni-hamburg.de

Abstract: Agent-based simulation can be viewed as methodical advancement and generalization of microscopic modelling styles in object-oriented and discrete event simulation well suited for logistic systems. Existing object-oriented simulation frameworks form an adequate basis for agent-based modelling where general simulation functionality can be reused. Extensions are required in view of the main aspects in agent-based simulation related to spatial and behaviour modelling. This paper outlines the advancement of a discrete event simulation framework for applications of agent-based simulation mainly in logistics. Agent-based simulation is typically applied in microscopic modelling of systems where common actions of autonomously deciding actors, mostly humans are represented. The extension includes a graph-based spatial modeling as well as flexible behavior descriptions in different modes (e.g. UML state charts). As an application of agent-based simulation and the related software tool in the logistics domain a simulation model for sustainable logistic concepts in courier services is presented.

1 Introduction

In computer simulation we generally carry out experiments with models, i.e. abstract mappings of the systems under investigation. Within the limits of model validity we can draw conclusions from our observations during a simulation experiment on the expected real system behavior under comparable conditions [PK05]. There is a large number of different modelling styles in computer simulation at hand depending in applicability from the problem of interest and the characteristics of the real system under investigation. Traditionally very often macro models are considered which are based on systems of differential equations and describe associations between variables on the level of the total system (i.e. traffic density and flow). However, microscopic models are more flexible and show a higher degree of model detail where the entities found in the real system are represented with their individual attributes and behavior (e.g. individual vehicles in a logistic model). Since several years more and more so-called agent-based simulation models (also termed as multi-agent models) are applied for microscopic modeling of systems whose behavior is characterized by common actions of autonomously deciding actors, mostly humans. The microscopic, mostly individual based perspective is characteristic for agent-based simulation where the analyzed system is modeled from the point of view of the involved actors.

This results in a high conceptual proximity between model and real system, because state and behavior of real world agents are represented directly in the agents of the model. Agent-based models feature a microscopic view on the simulated system. Actors present in the original real system are represented in the model as separate entities with an own identity, state and behavior. The behavior of the total system results from the interactions of the entities as macro phenomenon. This view also characterizes other modelling styles such as individual based, object oriented or discrete event simulation. Therefore a distinction between these different concepts is not straightforward.

An important characteristic of agent-based simulation beyond the typical spatial reference is the huge methodological variety and high abstraction level of the behavior description which have to meet characteristics such as autonomy, reactivity, communication and goal orientation ([PK05]). The concept of an agent is more general than the one of an individual, object or simulation process. While individual-based models usually consist of atomic units with simple, recurring behavior repertoire, different agents in a model can require heterogeneous forms of behaviour description. Established approaches in discrete event simulation such as process orientation are only suitable to a limited extend for the presentation of agents: Communication is restricted to a direct synchronization by mutual reactivation or indirect synchronization by applying for limited resources. Mostly linear processes are assumed where reactions on asynchronously occurring events are the exception.

The methodological variety of agent-based behavior modelling is reflected in agent-based software of very different kind where the spectrum is reaching from minor changes of object-oriented software concepts, over automata- or Petri-net based software development tools up to declarative modelling languages. While many software systems come from a time controlled advancement of a simulation clock, event control is a more efficient alternative, if the modelling of variable time steps is predominant. Agents become only active and carry out operational decisions if the entry of an external or a self-induced event is requiring this. In doing so time consumption has to be modelled explicitly by specifying reactivation points or by time intervals related to the durations of agent activity. Because of the conceptual closeness of the agent and object term object-oriented discrete-event simulation systems form a suitable basis for agent-based modeling and simulation. Existing general and model specific simulation functionality can be reused while extensions concerning space and behavior modelling can be embedded.

2 The Software Basis: An Object-Oriented Simulation Framework for Discrete Event Systems

A modeling tool called DESMO-J has been introduced as a simulation framework for discrete event simulation in Java several years ago (see [LP99], [PK05]) and applied regularly in simulation courses, in research and in technology transfer projects with industry as a suitable simulation tool, many from a logistic domain (e.g. harbor simulations [PN03], high rack warehouses or harbor backland traffic [Wi07]).

DESMO-J supports the event-oriented and process-oriented modelling style in discrete event simulation by abstract Java classes for entities (i.e. simulation objects), events and simulation processes (i.e. representations of the life cycle of the simulation objects) which can be extended model specifically [PK05]. Event- and process-oriented behavior descriptions are realized by redefinition of abstract methods for event routines or process life cycles, respectively. The implementation of the process-oriented simulation approach requires a co-routine control which is based on Java threads synchronized with the simulation control and result in a much higher run time as compared to event-oriented simulation software. DESMO-J is public domain software and licensed under the APACHE public license (see www.desmoj.de). The framework provides the user with the standard functionality for discrete event models and provides some predefined abstract classes where the user can add his model-specific behavior to the framework (so-called Hot-Spots). In summary the simulation functionality of DESMO-J includes:

- The simulation scheduler encapsulating the event list and simulation clock;
- process-oriented as well as event-oriented simulation objects – including their combination;
- queues with priorities and automatic statistics;
- replaceable random number generators and various statistical distributions (Uniform-, Exponential-, Normal-, Erlang-, Poisson-, Triangle, empirical distribution, etc.);
- higher modelling components for process synchronization, i.e. more compact simulation modeling on a higher abstraction level such as resource competition, conditional waiting, direct process cooperation (Master/ Slave relations), storage bins as inventories, or process interrupts;
- automatic reporting for queues, distributions and higher model components;
- reporting, trace, debug and error output in HTML and XML format (i.e. platform independent representation of output data).

An important asset of DESMO-J is the *separation* of *models* from *experiments*, i.e. model development is kept independently from simulation experiments (defined in a so-called *experimental frame* – a well established paradigm in discrete event simulation). We achieve this by differentiation of model- (or model component-) objects of `class Model Component` from experiment objects of `class Experiment` (with the specification of the experimental parameters).

To run a simulation with a given model a model object has to be instantiated from the user model class (as derived from abstract `class Model`) and connected to the experiment object. As one advantage we can connect several experiment objects with one model object and run this experiment series without program restart. Beyond that this separation of models and experiments is required to allow for reuse of (sub-) models as components in other models on a higher level. In this way we introduce the basis for a submodel concept in DESMO-J where each model consists of a set of submodels providing model reusability and complete model hierarchies.

3 Extension of the Simulation Framework by Agent-Based Concepts

The Framework FAMOS for agent-based simulation extends the discrete event simulator DESMO-J by agent-based concepts of spatial and behaviour modelling. The constructs for spatial modeling, which general idea has been introduced and outlined in [Me01], also support directional graphs and continuous spaces beyond the common regular grids. Graph based models are for instance useful in the context of logistic modelling in order to represent road networks. Agents move through space following different specified and extensible strategies, e.g. randomly or along gradient arrays and previously determined routes.

For the integration of simulated agents into the discrete event simulation framework DESMO-J the event-oriented functions have been chosen mainly for efficiency reasons. User defined agents can be derived in FAMOS from the base `class Agent`, which itself is an extension of `class Entity` for entities in event-oriented models. Entities are generally passive in DESMO-J. Its state is altered externally by model specific event routines derived from base `class Event`. An agent is encapsulating a state as well as a dynamic behavior and processes actively occurring events conveying information on externally triggered system state alterations as well as self-induced state changes.

Thus FAMOS-agents communicate with their environment by typed signals represented by objects of subclasses of abstract `class Signal` and indicating changes of model states. Signals are defined by their type and a list of parameters a user has to declare by derivation of a subclass. The agent receives signals either from its environment or schedules them himself as intended future activities. By receiving a signal an activation entity derived from `class Event` is entered into the event list of DESMO-J activating the agent for processing it. The activity selection by the agent is effected by reaction on signals and is delegated to a proxy object of `class Behavior` encapsulating the agent behavior.

Several approaches for behavior modeling have been realized in FAMOS including state charts with UML semantics. The FAMOS state chart interpreter is an extended version of the state chart framework from the open source UML case tool Fujaba [Kö99]. State charts seem to be very suitable for modelling sub-cognitive agents, because its graphical clearness allows for a straightforward verification of agent behavior. Graphical specifications can be directly mapped on operational agents using code generators and visible programming tools allowing for a considerable reduction of the critical gap between conceptual model and implementation in simulation modelling. For this reason FAMOS incorporates and a graphical state chart editor (Fig. 1).

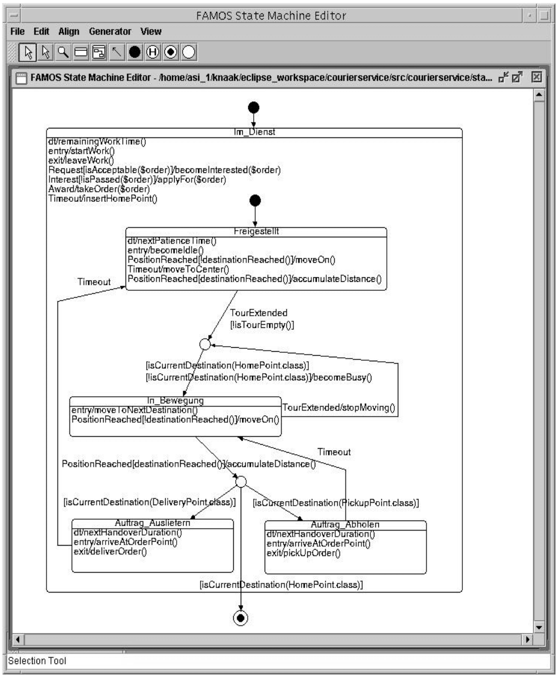

Fig. 1: Graphical editor for modelling agent behaviour with UML state charts. The example shows the hierarchical state chart of a simulated city courier.

The behavior description with state charts has been applied for the construction of an agent-based model for city courier briefly discussed in the next section. But there are several other possible applications. Another example in the logistic domain is a prototype simulator for boarding Airbus jets [Cz03].

4 Application of the Agent-Based Simulation Tool in City Courier Logistics

As an application example of our agent-based simulation tool a case study from the logistics domain is briefly introduced. The case study deals with simulation of sustainable logistic systems for city courier services. Courier services are an important factor of product and business traffic in many large cities. As compared to express services the individual delivery of consignments by couriers is leading to a higher flexibility on one hand. On the other hand it causes a high traffic density, which is critical from the economic as well as from the ecological side.

In a funded research project an agent-based simulation model of city courier services has been developed in cooperation with two local courier firms. By means of this model ecological, economic as well as social implications of two alternative logistic strategies have been analyzed and evaluated. Simulations were run to compare the existent organization of typical courier services with two new logistic concepts based on hubs.

The agent-based view for simulation modelling seems appropriate for modelling courier services mainly for three reasons:

- Courier services own only a limited degree of central control. Couriers are acting largely autonomously since they decide independently on the acceptance of deliveries and their tour planning. It seems obvious to model the activities in a courier service from the viewpoint of the involved actors (couriers and radio operators).

- The couriers are moving through a spatial environment, i.e. the road network. They follow independently planned routes (using a modified version of Dijkstra's shortest path algorithm [DD02] that considers averaged travelling durations based on speed limits of different road types) which have to be modified during the tour reacting on the exclamation of new deliveries by the radio operator. This balance of reactive and proactive behavior is a typical characteristic of agents.

- The switching protocol applied in courier services is a type of the contract net protocol for the distributed allocation of tasks known from distributed artificial intelligence as shown in Fig. 2 ([Sm80]).

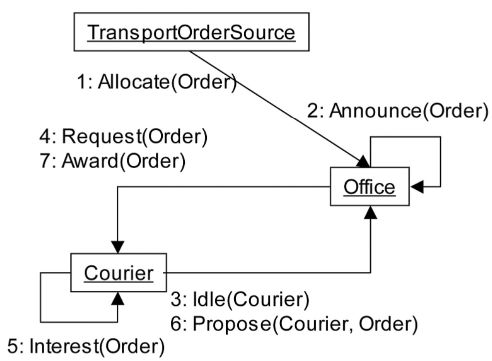

Figure 2: UML collaboration diagram showing the contract-net like mediation protocol of the agent-based courier service model

Our approach for optimizing city courier services is the enhancement of the courier concept by Hubs known from express services. These logistic strategies (see Fig. 3) together with an increased employment of bicycle couriers and bundling of consignments seem to be a sensible way for reducing costs, transportation distances, and emissions.

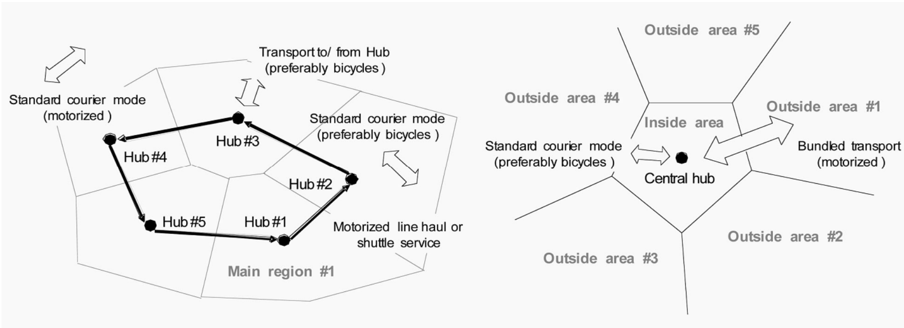

Figure 3: Alternative logistic concepts: Hub and Shuttle (left) and Inside/Outside (right)

Concerning ecological measures like the total motorized distance the Inside/Outside strategy displayed a slight improvement compared to the status quo. Promising results have been found in the simulation studies by introducing a single inner city interchange station for bundling deliveries for the suburban areas (for more details, see [Kn04], [Kn06]). A comparison of the total motorised distance covered by couriers in the 3 different models per work day is shown in Fig. 4. This comparison is based on 5 different order profiles, e.g. historical data sets of transport orders provided by our cooperation partners.

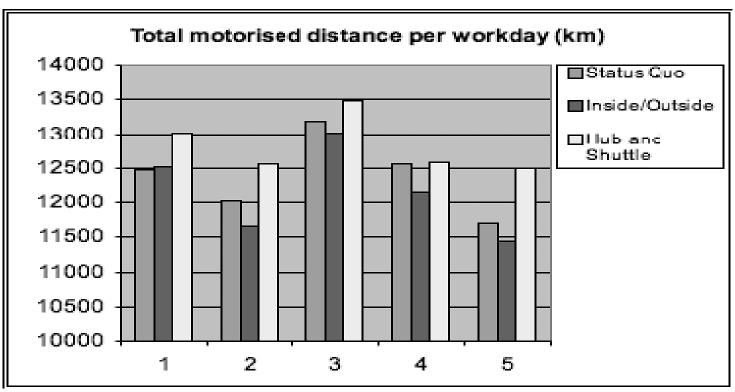

Figure 4: Comparison of the total motorised distance covered by couriers

5 Conclusions

In this paper we presented an extension of a discrete event simulation framework in Java for modelling agent-based systems. With this simulation framework an agent-based model of city courier services has been designed and implemented. The aim of the simulation study was the analysis of alternative logistic structures which allow for a sustainable operation of courier services. The agent-based modelling approach supported by adequate modelling tools is well suited for logistic services which operate with autonomous actors (e.g. couriers) in a spatial environment.

References

[Cz03] Czogalla, R., Meyer, R., Page, B., Ciernioch, G.: Entwicklung eines Werkzeuges zur agentenbasierten Simulation von Personenbewegungen innerhalb einer Flugzeugkabine. Proc. Simulationstechnik – 17th Symposium, Magdeburg, Sept. 2003, SCS, Delft 2003, pp. 137-142.

[DD02] Domschke, R., Drexel, R.: Eine Einführung in Operations Research. Springer, 2002.

[De03] Deecke, H., Knaak, N., Meyer, R., Page, B.: Nachhaltige Logistikkonzepte für Stadtkurierdienste, Technical report, University of Hamburg, 2003.

[Dr02] Drogoul, A., Vanbergue, D., Meurisse, T.: Multi-Agent Based Simulation : Where are the Agents. In: Sichman et al.: Multi-Agent Based Simulation II, Springer, Berlin, 2002, pp. 2-15.

[Fe99] Ferber, J. : Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence. Addison-Wesley, Harlow, 1999

[Hi01] Hilty, L. M., Meyer, R., Ruddy, T. F.: A General Modelling and Simulation System for Sustainability Impact Assessment in the Field of Traffic and Logistics. In: Rautenstrauch, C., Patig, S. (eds).: Environmental Information Systems in Industry and Public Administration. Idea Group Publishing, Hershey: Pa, USA, 2001.

[Kl01] Klügl, F.: Multiagentensimulation – Konzepte, Werkzeuge und Anwendungen. Reading, MA.: Addison-Wesley, 2001

[Kn04] Knaak, N., Page, B., Meyer, R.: Logistic Strategies for Sustainable City Courier Services – An Agent-based Simulation Approach. In: A.G. Bruzzone, et al. (eds.), Proc. International Workshop on Harbour, Maritime & Multimodal Logistics Modelling and Simulation – HMS 2004, Rio, Brazil, Sept. 2004, pp.139-144.

[Kn06] Knaak, N., Kruse, S., Page, B.: Agent-based Simulation for Modelling Sustainable Logistics Systems. In: Voinov, A., Jakeman, A., Rizzoli, A. (Eds): Proc. of the IEMSS 3rd Biennial Meeting "Summit on Environmental Modelling and Software". Burlington, USA, July 2006. CD ROM

[Kö99] Köhler, H.-J.: Code-Generierung für UML Kollaborations-, Sequenz-, und Statechart-Diagramme, diploma thesis, University of Paderborn. Paderborn, 1999.

[LP99] Lechler, T., Page, B.: DESMO-J – An Object-Oriented Discrete Simulation Framework in Java. In: Proc. Simulation in Industry '99 – 11th European Simulation Symp. 99, Erlangen, SCS Publ., Delft, Netherlands. 1999, pp. 119-124.

[Me01] Meyer, R.: Bewegungsgraphen – Ein Konzept für die Modellierung der Umgebung in der individuen- und agentenbasierten Simulation. Proc.Workshop Simulation in Umwelt- und Geowissenschaften; Münster, March 2001, Shaker, Aachen, 2001. pp. 47-60.

[PN03] Page, B., Neufeld, E.: Extending an object oriented Discrete Event Simulation Framework in Java for Harbour Logistics. Intern. Workshop on Harbour, Maritime & Multimodal Logistics Modelling and Simulation – HMS 2003, Riga, Letvia, Sept. 2003, pp. 79-85.

[PK05] Page, B., Kreutzer, W.: The Java Simulation Handbook. Simulating discrete Event Systems in UML and Java. Shaker Publ., Aachen 2005.

[Sm80] Smith, R. G.: The Contract net Protocol: High Level Communication and Control in a Distributed Problem Solver. IEEE Transactions on Computers 29 (12), 1980, pp.1104-1113.

[Wi07] Wittmann, J., Göbel, J., Möller, D., Schroer, B.: Refinement of the Virtual Intermodal Transportation System (VITS) and Adoption for Metropolitan Area Traffic Simulation. Submitted to the 2007 Summer Computer Simulation Conference (SCSC'07), San Diego, USA, July 2007