

AGENT-BASED AND DISCRETE EVENT SIMULATION OF AUTONOMOUS LOGISTIC PROCESSES

Markus Becker	Jan D. Gehrke
Bernd-Ludwig Wenning	Martin Lorenz
Carmelita Görg	Otthein Herzog
TZI - Communication Networks and CRC 637	TZI - Intelligent Systems and CRC 637
Otto-Hahn-Allee NW1	Am Fallturm 1
University of Bremen	University of Bremen
28359 Bremen, Germany	28359 Bremen, Germany
{mab wenn cg}@comnets.uni-bremen.de	{jgehrke mlo herzog}@tzi.de

Keywords—Simulation, Agent-based, Discrete Event, Logistics, Transports, Communication

Abstract—The current trends and recent changes in logistics lead to new, complex and partially conflicting requirements on logistic planning and control systems. Currently available strategies and methodologies do not address these new requirements sufficiently. The concept of autonomous logistic processes intends to overcome these drawbacks together with latest information and communication technologies. Their analysis and design is subject to simulation studies. Two simulation systems for the analysis of autonomy in logistics with an agent-based and a discrete event approach are presented. Both systems are designed and suitable for different aspects of autonomous logistic processes.

I. INTRODUCTION

Logistics planning and monitoring poses a technically challenging problem. The various types of planning activities include packing, scheduling, and route planning with several constraints. The problem is highly distributed, yet interconnected (as the organisations are autonomous). Dynamic situations enter the system, e.g., in the form of changed logistics requests. The simplest plans involve interaction between the components that are distributed in the logistic networks.

Discrete event simulation (DES) has been widely used for the simulation of communication networks and transport networks. Wenning et al. (2005) showed simulation results of adaptive reactive routing protocols in transport networks, while in (Becker et al., 2005) the combined simulation of communication and transportation networks was shown with respect to programming features of object-oriented languages like templates, access control, and design patterns.

Agents represent a state-of-the-art approach for implementing autonomous systems. A multiagent system is a loose aggregation or society of agents each with clearly defined roles, responsibilities and functionality. The smallest controlling entity in this approach, an agent, is commonly described as anything that is able to 'perceive its environment through sensors and act upon that environ-

ment through actuators' (Russell and Norvig, 2003, p. 32). The system architecture considered, a *multiagent system* (MAS), is one that consists of a number of agents, which interact with one another. Hereby a perceivable common behaviour emerges.

A recent development in multiagent systems research is their application in transportation and production logistics (Klügl et al., 2005; Dorer and Calisti, 2005; Denkena et al., 2003). In the TeleTruck approach Bürckert et al. (2000) describe a fleet scheduling system based on their concept of a holonic MAS. Hofmann et al. (1999) concentrate on the information exchange for tracking logistic entities. They developed a decentralised agent-based search-on-demand information infrastructure as a more flexible replacement of EDIFACT-based pushing of information. The focus of Moyaux et al. (2003) is on the supply chain and the well-known bullwhip effect. They propose a MAS-based coordination technique to reduce the fluctuations of orders.

The remainder of this paper is structured as follows: Section II gives an overview of research on autonomous logistic processes. In Section III two simulation systems for autonomy in logistics with different purposes, one using an agent-based and the other a discrete event approach, are presented. Both systems are compared in Section IV. The paper concludes with a summary and plans for further research (Sec. V).

II. AUTONOMOUS LOGISTIC PROCESSES

The dynamic nature of modern transport networks increases the complexity of decision-making in today's logistics. An exact or even heuristics-based solution for global optimisation becomes impossible. Since the distribution of planning and decision-making to autonomous components is a widely accepted promising solution to handle complex problems (Jennings, 2001), we consider it an appropriate set-up for logistic systems.

We assume that both plans and reactions to changing conditions will not be calculated on some central unit in the network, but decentralised in those units which are

primarily affected (e.g., transportation hubs, vehicles or even packages). This decentralised approach implies that the unit which decides about its next steps might only have limited knowledge about its environment.

A logistics system based on the above principles allows the transfer of more decision competence from the logistics service provider to autonomous representatives of the logistics system user (which, in a far reaching scenario, may be a single package agent). Furthermore, it creates a new approach to routing and mode-choice problems. Considering packages as autonomous, we may focus on transferring models and routing algorithms from the network layer of communication networks to logistics. The investigation of algorithms is supposed to include different approaches from artificial intelligence, operations research, and communication networks.

In logistics research an ongoing paradigm shift from centralised control of non-intelligent items in hierarchical structures towards decentralised control of intelligent items in heterarchical structures specially in logistic processes can be observed. Those intelligent items can be all sorts of logistic objects from raw materials, components or products to transit equipment (e.g., pallets, packages) or transportation systems (e.g., conveyors, trucks) (Scholz-Reiter et al., 2004).

III. SIMULATION OF AUTONOMOUS LOGISTIC PROCESSES

As one of the primary goals of the common work is the development of concepts for autonomous logistic processes, there has to be a way to check these concepts for feasibility and performance. This can be achieved by modelling the processes in a proper way and creating and running simulations with logistic scenarios that represent a logistic world with integrated concepts for autonomous components. From the simulation results, conclusions can be drawn concerning the feasibility of the concepts, and comparisons can be made between different new approaches and current logistic processes.

One of the approaches to be investigated for autonomous logistic processes is the idea to transfer and adapt routing methods from communication networks to routing of autonomous logistic components. Following this approach, it is not far-fetched to also use the simulation methods utilised when doing research on communication networks. In the communication networks community, a widely used method to qualitatively and quantitatively investigate protocols is the *discrete event simulation* (DES). There are a couple of simulation tools and libraries available for this. The CNCL (Görg et al., 1993), a DES class library, has been used as a basis to create a simulation environment for autonomous logistic processes.

Moreover, the cooperation of autonomous logistic objects implies that these objects have to communicate. Different concepts of cooperation also differ in where or when communication is required and what data volume has to be transmitted. Therefore, it is necessary to investigate the communication requirements of an approach in order to validate its relevance for practical use. Due to this rea-

son, the DES is not only able to model the logistic components, but also their communication. This makes the DES environment capable of showing both the influence of the cooperation concepts to the communication and vice versa.

Agent technology is a promising field to fit in the distributed nature of the logistic network paradigm, due to the widely distributed heterogeneous open nature of agent platforms. It has the potential to play a key role in building and supporting virtual enterprises, enriching higher level of communication and enabling more intelligent service provisioning. At the same time multiagent-based simulation has become an important approach in various domains. It is obvious to evaluate and compare the different agents' behaviours and performance in an agent-based simulation before they are deployed in real-world logistics.

A. Agent-based Simulation

As a generic platform for simulating and evaluating logistic scenarios with logistic entities as autonomous actors an agent-based simulation system has been developed. This system allows for a flexible mapping of logistic entities to software agents and provides logging, evaluation, introspection, visualisation, and interaction features. In general, the simulation platform is designed to support arbitrary logistic scenarios from logistics in transportation and production, including multi-modal transports and scenarios that cover the complete supply chain. However, the current simulation scenarios are focused on transportation logistics and road traffic, i.e., agents are chiefly representatives of trucks or their load.

A.1 Architecture and Control

The platform is designed upon the FIPA-compliant Java agent development framework JADE (Bellifemine et al., 2003). It supports the distribution of the simulated logistic entities to multiple computers. The top-level simulation control instance is an agent called *SimulationManager*. It is responsible for the handling of simulation state transitions (e.g., starting, pausing, stopping), the set-up of the scenario and its agents including distribution to multiple computers, and agent synchronisation. For each computer configured to participate in the simulation (cf. III-A.2) the *SimulationManager* starts a sub-controller agent, called *ContainerManager*, on the respective machine. This sub-controller is the primary contact point when sending (broadcast) control messages and "world events" (i.e., events occurring in the modelled world) to the agents in the simulated scenario. The sub-controller forwards these messages to the agents on its computer thereby avoiding cross-platform communication traffic.

The agent-based simulation is synchronised by sending events for discrete steps. These steps have a fixed but configurable length in model time. The simulation run-time for one step varies depending on the calculations needed for an agents' actions during one step.

A.2 Configuration

The configuration of an agent-based logistics simulation on the platform incorporates different aspects:

1. the initial state of the logistics scenario,
2. the mapping of logistic entities to software agent classes,
3. the computers that host the distributed simulation,
4. the visualisation of different maps and logistic entities,
5. the human users' control permissions,
6. the logging properties, and
7. other control attributes.

The initial state includes the participating and autonomous logistic entities (e.g., suppliers, customers, and means of transport) and their respective types, properties, and capabilities. Another important part of the scenario specification is the transport network which consists of a directed graph with typed and annotated nodes and edges. The scenario is specified in a formal, machine-processable way using an ontology in the W3C standard web ontology language OWL (McGuinness and van Harmelen, 2004) in its OWL DL sub-language. In order to specify a scenario one uses existing domain description ontologies (T-Box in description logic terminology) and states instances of classes in these domain descriptions with their respective properties in a new scenario ontology (A-Box) that imports all necessary domain ontologies.

Background knowledge on the logistics domain is described in five ontologies. The first is the general ontology containing concepts and concept relations common for most scenarios and sub-domains in logistics. The other four ontologies are extensions of the upper ontology for a special sub-domain. These ontologies are separated in *transportation*, *production*, *information and communication technology*, and *goods*. The ontologies not only define the initial scenario state, they provide a common formal language for the agents to represent and communicate the state of the simulated world.

A.3 Agents and Agent Interactions

Agents in the logistics simulator may be, e.g., suppliers, customers, means of transport, transshipping facilities, traffic nodes, cargo that is shipped, and services like routing, traffic information or brokers for transport capacity offers and transport requests. The agents communicate by passing messages in the FIPA agent communication language ACL and coordinate using FIPA interaction protocols (IPs), e.g., the contract net IP. The incoming ACL messages are filtered and handled by multiple agent behaviours that are scheduled by JADE.

To be able to use the agent communication features provided by JADE, additional communication ontologies for some services and agent interactions have been defined. There are ontologies for communications in (1) transport mediation and processing, (2) routing, and (3) sensor monitoring. These ontologies are expressed in the FIPA Semantic Language SL0. Messages from the simulation controller and sub-controllers are also defined in a FIPA SL0 communication ontology (Foundation for Intelligent Physical Agents, 2002).

A.4 Scenario Visualisation and Analysis

The agents and simulation controllers may log any desirable logistics metrics, e.g., fill rate, inventory, cost, cy-

cle time, or other information, e.g., their current position, route, accepted orders, or detected incidents. This data is stored in a relational data base for later analysis or live monitoring as done by the visualisation tool depicted in Fig. 1. This viewer enables a comprehensible presentation of the scenario and the agents' behaviours. The simulation controller is able to decelerate or pause the simulation if needed.

B. Discrete Event Simulation

B.1 Development Basis and Architecture

The DES environment was developed based on the *Communication Networks Class Library* (CNCL), a class library developed at RWTH Aachen for the simulation of communication networks (Görg et al., 1993). This class library provides all the basics that are needed for event-based simulation like scheduling, event handling, random generators, statistical analysis as well as other general classes. The major components of a CNCL-based simulation are the scheduler and the event handlers. The event-processing simulation objects have to be modelled as event handlers derived from a generic event handler.

In the simulation, the objects send events to the scheduler which include information about the event's destination, the time the event has to be fired and in some cases objects that are attached to the event. When the time for the event is reached, the scheduler forwards it to its destination. In the object receiving the event, the event handling method executes the object's actions depending on the content of the event. So the logistic objects are modelled as object-oriented objects with event handlers. The autonomy of the logistic objects is implemented in the event handling method.

B.2 Logistic Model

The DES model is based on a scenario description. This scenario description is composed of several instances of the following transport related components:

Vertices are locations in the network with either a possibility of direction change or the possibility for transshipment or both.

Sources are extensions to vertices, "generating" packages. Details of the generation (package destinations, generation rates) are also specified in the scenario.

Sinks are vertices to which packages are delivered. Explicit specification of sinks can be omitted as the source details already include implicit definition of sinks.

Edges are connections between vertices. Edges are considered to be directed.

Vehicles are transport units moving on the edges and carrying packages.

Packages are the goods to be transported. A package is considered to be indivisible.

Furthermore, the scenario contains the following communication relevant objects:

CommunicationUnit is the device enabling the logistic objects to communicate. A logistic object may have multiple CommunicationUnits for different communication networks.

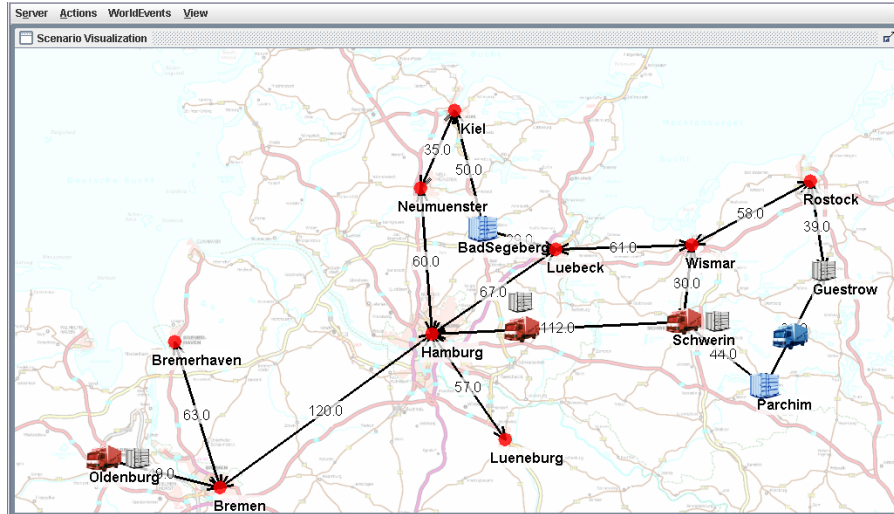


Fig. 1. Scenario viewer UI showing a small scenario with a map of northern Germany, trucks shipping containers, and waiting containers.

CommunicationUnitManager manages multiple *CommunicationUnits* and selects one, according to certain requirements.

MetaCommunicationUnitManager performs the communication and enables tracking of the amount of communicated data.

Most of the logistic and communication components mentioned here have a couple of attributes, e.g. a vehicle has a capacity, a maximum speed and others. The scenario description is given in XML formatted input files. At the initialisation of a simulation, these input files are parsed and all objects contained in the scenario are created and initialised with the respective attributes.

A sample scenario consisting of 18 German cities as vertices and the freeways interconnecting them as edges is depicted in Fig. 2.

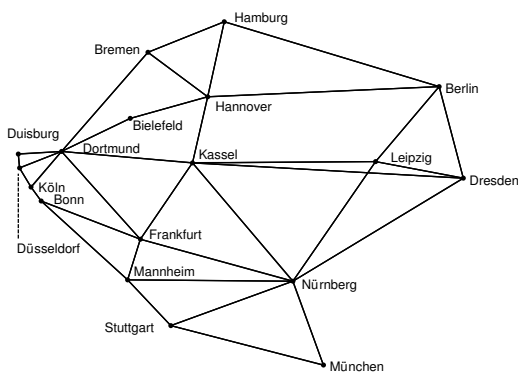


Fig. 2. Sample Scenario for discrete event simulation

The simulation is specific to the interrelated communication and transport problem. It is neither a simulation of just transport nor a simulation of just communication, but an interconnected simulation, where each autonomous logistic object acts on its own and its communication need is recorded.

The output observed in the DES is the packet travel time, utilisation of the vehicles, etc. in the format of histogram and probability density function. Furthermore the

routes taken by packages and vehicles are recorded. The total amount of data communicated, amount of data per vehicle, per communication system etc. is output with respect to the communication.

IV. COMPARISON OF THE CONCEPTS

The described systems share the goal of simulating aspects of autonomous logistic processes. However, they differ in a variety of important aspects because they were designed for different purposes and application areas.

In the following the two concepts, i.e., agent-based and discrete event simulation, are juxtaposed regarding a number of attributes and it is shown where the two concepts are similar, differ or extend each other.

A. Flexibility

The agent-based simulation system primarily aims at evaluation and analysis of intelligent agents and the mechanisms implementing their intelligence. It is a testbed for different approaches to autonomous processes implemented as agents, e.g., for logistic entities. Thus, it sets a high value on flexibility in configuration of scenarios and their actors as well as graphically supported comprehensibility and traceability of the actors' behaviours. Furthermore, the MAS programming paradigm fits nicely the idea of autonomous entities representing logistic objects. At the same time it allows for an easy adaptation of agents tested in simulations for real-world applications. This has already been shown in a small demonstration setting.

The DES on the other hand is very specifically designed to the problem of simulating autonomous logistic processes. In the event-based simulation the autonomy is hard to grasp, as it is closely integrated into the object-oriented classes. The event-based simulation in addition has to reimplement interaction protocols each time it uses them, while the agent-based simulation has this feature available built into the MAS. Additionally it is hard to have different behaviours, e.g., for different shippers, in the DES, whereas in the agent-based simulation, the behaviours are encapsulated and are easily exchangeable.

Concerning the level of detail of modelling, both the discrete event and the agent-based simulation are very similar. There are no hard limitations for how detailed the scenarios can be, higher level of detail just means more events and maybe additional objects in the DES whereas it means additional messaging and presumably additional agents in the agent-based environment. The only practical limit in both cases is the feasibility of the simulation in terms of memory and processing power requirements, i.e. limitations due to the hardware that runs the simulation.

B. Time concepts

The stepwise simulation process is a possible drawback of the current agent-based system because it may complicate the transfer to real-world applications and causes an administration overhead in large-scale simulations.

The discrete event simulation handles the time in such a way that during an event no model time passes, while computation time is consumed. Between events the model time passes by, but almost no computation time is used, except for event scheduling purposes.

Real simultaneousness is a problematic issue in both types of simulation. If, for example, two vehicles reach the same vertex simultaneously, possible transshipments between them depend on which one of them is processed first in the simulation. In the MAS, it depends on the order in which the threads are processed in a time step, whereas in the DES, it depends on the order of the events in the schedule.

C. Structure

The structural components of the two simulation components are closely interrelated with each other. For example the ACL message of the agent-based simulation serves the same purpose as the Event class in the DES. The agents' Behaviour class is comparable to the EventHandler in DES.

The world model underlying the agent based simulation provides an explicit semantic and enables higher level reasoning capabilities within and communication among autonomous entities.

D. Reproducibility and Statistical Evaluation

The results of the DES are always reproducible. If the same simulation is started with the same seed for the random number generators, it always produces the same results. In agent-based simulation, the concurrency of agents leads to an implicit random component that makes each simulation run unique. The concurrency that is present in MAS cannot be introduced into DES, as this would affect the reproducibility.

For the discrete event simulation, methods for detailed statistical evaluation can be taken from the CNCL library. These evaluation methods provide the possibility to give reliable statistical results as they implement proven methods like batch-means or *limited relative error* (LRE) evaluation.

Parts of the CNCL, namely the *Random Number Generators* (RNGs), the distributions, and the statistical evaluation classes have already been ported to Java in order to complement the MAS. The resulting port is called jCNCL.

E. Runtime Behaviour

Regarding the runtime performance, a significant difference between the simulation concepts can be observed. The DES simulation has been observed to be nearly 400 times faster than the agent-based simulation in a scenario with 25 nodes arranged in a regular square grid and 250 homogeneously distributed vehicles. This can be seen as the price that has to be paid for the greater flexibility and better representation of reality in the agent-based system. For example, while the DES objects send raw data in events and method invocations, the MAS encapsulates all data which has to be communicated into ACL messages normally including String parsing. This slows down the simulation, but is closer to real applications. In addition the JADE *directory facilitator* (DF) agent has been identified as a major bottleneck. The runtime performance in the MAS simulation system has been paid less attention yet. First analysis has shown high potentials for improvements here, especially by reducing multi-threaded behaviours within agents. In order to speed-up multiagent simulations, Grid technologies have also been evaluated (Timm and Pawlaszczyk, 2005).

V. CONCLUSIONS AND FURTHER RESEARCH

In this paper two simulation systems for analysis of autonomous logistic processes were presented. The properties of the two systems are summarised in Tab. I.

Criteria	DES	MAS
Flexibility	–	+
Autonomy Encapsulation	–	+
Reproducibility	+	–
Statistical Evaluation	+	+
Runtime Performance	+	–

TABLE I: Summary of properties

The distributed multiagent simulation system provides a flexible testbed to analyse and compare different algorithms for autonomy in complex environments and also features a scenario visualisation component. Its major advantage is that it perfectly meets the notion of autonomous logistic entities that directly interact with each other. Thus it is supposed to be easily transferable for real-world applications.

The presented discrete event simulation system is more specific and thereby it is particularly well suited for the analysis of transportation and communication activities in autonomous logistic processes. Moreover, in contrast to the agent-based approach, the discrete event simulations are reproducible on the computational level and thus provide an authoritative testing environment.

Despite their different purposes, in particular the investigation of communication requirements, it is intended to integrate the route planning and also the communication analysis into the agent-based simulation. As a first step to improve system comparability a converter for both scenario specifications for the other system respectively will be implemented.

As mentioned above, a second goal is to accelerate the stepwise simulation process in the agent-based simulation environment. In cooperation with TU Ilmenau a distributed, event-based control with optimistic synchronisation is implemented. Optimistic synchronisation approaches promise an improvement in performance by allowing agents to diverge in time and synchronise them by roll-back only if both partial simulations interfere with each other.

ACKNOWLEDGEMENT

This research was supported by the German Research Foundation (DFG) as part of the Collaborative Research Centre 637 "Autonomous Cooperating Logistic Processes".

REFERENCES

- Becker, M., Wenning, B.-L., and Görg, C. 2005. Simulation of Communication Networks and Logistical Networks – Using Object Oriented Programming Language Features to Enhance Modelling. In *Symposium on Modeling and Simulation Tools for Emerging Telecommunications Networks - Needs, Trends, Challenges, Solutions*.
- Bellifemine, F., Caire, G., Poggi, A., and Rimassa, G. 2003. Jade - a white paper. online publication.
- Bürckert, H., Fischer, K., and Vierke, G. 2000. Holonic transport scheduling with teletruck. *Journal of Applied Artificial Intelligence*, 14:697–725.
- Denkena, B., Woelk, P.-O., Herzog, O., and Scholz, T. 2003. Integration of process planning and scheduling using intelligent software agents. In *Proceedings of the 36th CIRP-International Seminar on Manufacturing Systems*, Saarbruecken, Germany.
- Dorer, K. and Calisti, M. 2005. An adaptive solution to dynamic transport optimization. In *AAMAS '05: Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pages 45–51. ACM Press, New York, NY, USA. ISBN 1-59593-093-0.
- Foundation for Intelligent Physical Agents 2002. Fipa sl content language specification. online publication.
- Görg, C., Junius, M., and Walke, B. 1993. *Hand-on-Tools Praktikum*. RWTH Aachen, Germany. Available at <http://www.comnets.rwth-aachen.de/doc/cncl.html>.
- Hofmann, O., Deschner, D., Reinheimer, S., and Bodendorf, F. 1999. Agent-supported information retrieval in the logistics chain. In *HICSS '99: Proceedings of the Thirty-second Annual Hawaii International Conference on System Sciences-Volume 8*, page 8028.
- Jennings, N. R. 2001. An agent-based approach for building complex software systems. *Comms. of the ACM*, 44(4):35–41.
- Klügl, F., Bazzan, A., and Ossowski, S., editors 2005. *Applications of Agent Technology in Traffic and Transportation*, volume VIII. Birkhäuser.
- McGuinness, D. L. and van Harmelen, F. 2004. OWL web ontology language overview.
- Moyaux, T., Chaib-draa, B., and D'Amours, S. 2003. Multi-agent coordination based on tokens: reduction of the bullwhip effect in a forest supply chain. In *Proceedings of the Second International Conference on Autonomous Agents & Multiagent Systems (AAMAS'03)*.
- Russell, S. J. and Norvig, P. 2003. *Artificial Intelligence: A Modern Approach*. Prentice Hall, New Jersey, 2 edition.
- Scholz-Reiter, B., Windt, K., and Freitag, M. 2004. Autonomous logistic processes: New demands and first approaches. In Monostori, L., editor, *Proceedings of the 37th CIRP International Seminar on Manufacturing Systems*, pages 357–362, Budapest, Hungary.
- Timm, I. J. and Pawlaszczyk, D. 2005. Large scale multiagent simulation on the grid. In Veit, D., editor, *Proceedings of the Workshop on Agent-based Grid Economics (AGE 2005) at the IEEE International Symposium on Cluster Computing and the Grid (CCGRID)*, Cardiff, UK.

Wenning, B.-L., Görg, C., and Peters, K. 2005. Ereignisdiskrete Modellierung von Selbststeuerung in Transportnetzen. *Industrie Management*, 5:53–56.

AUTHOR BIOGRAPHIES

MARKUS BECKER received his Diploma in Electrical Engineering and Information Technology in 2004 at the Aachen University of Technology, Germany. In September 2004 he joined the Communication Networks Group at the University of Bremen as a Research Assistant and a PhD candidate. Currently he is involved in the CRC 637. He is a member of COST 285 'Modelling and Simulation Tools for Research in Emerging Multi-service Telecommunications'. His research interests include Sensor and Ad-hoc Networking, Simulation of Communication Networks and Applications of Telecommunication Networks. His Web-page can be found at <http://www.comnets.uni-bremen.de/~mab/>.

BERND-LUDWIG WENNING received his Diploma degree in Electrical Engineering and Information Technology at the University of Bremen in 2002. Since then, he works as a Research Assistant and PhD candidate in the University's Communication Networks Group. Since the establishment of the CRC 637, his research focus is on the CRC subproject B1 "Reactive Planning and Control". His Web-page can be found at <http://www.comnets.uni-bremen.de/~wenn/>.

PROF. DR. CARMELITA GÖRG is leading the Communication Networks Group (ComNets) at the University of Bremen which is part of the tzi/ikom (Center for Computing Technologies / Center for Communication and Information Technology) and the MRC (Mobile Research Center). Prof. Görg has published a large number of scientific papers mainly in the field of communication networks. She is a speaker of the ITG working group 5.2.1 on "System Architecture and Traffic Engineering" and member of the board of the ITG (German Information Technology Society). Her research interests include: Performance Analysis of Communication Networks, Stochastic Simulation, Rare Event Simulation, High Speed Networks, Personal Communication, Wireless Networks, Mobility Support, and New Services in Telecommunication Networks. Her Web-page can be found at <http://www.comnets.uni-bremen.de/~cg/>.

JAN D. GEHRKE received his diploma degree in computer science from the University of Bremen in 2005. He graduated with a thesis on knowledge-based scene analysis for intelligent vehicles. In February 2005 he joined the artificial intelligence group at the University of Bremen as a Research Assistant and PhD candidate. He is affiliated to the CRC 637. His research focuses on intelligent agents in logistics as well as knowledge representation and management in multiagent systems. His Web-page can be found at <http://www.tzi.de/~jgehrke/>.

MARTIN LORENZ received his diploma degree in computer science from Technical University of Vienna. He graduated in 1999 with a thesis on 'Signing with the Computer - What User Interface Design can Learn from the Visual Language of Deaf People'. After two non-scientific affiliations from 2000 to 2003 he joined the artificial intelligence group at the University of Bremen in fall 2003 as a Research Assistant. He is now affiliated to the CRC 637. He currently aims at his PhD on autonomous decision making and risk management in rational agents. His Web-page can be found at <http://www.tzi.de/~mlo/>.

PROF. DR. OTTHEIN HERZOG is head of the Artificial Intelligence Research Group, which he built up when he was appointed as a full professor to the Computer Science Department of the University of Bremen in 1993. Before he joined academia, he worked in industrial software development with IBM for 16 years, where he collected extensive know-how in numerous international software product development and AI research projects. Dr. Herzog is the director of the Center for Computing Technologies (TZI) at the University of Bremen and director of the CRC 637. In addition he is co-founder and director of the Mobile Research Center (MRC) at the University of Bremen. His Web-page can be found at <http://www.tzi.de/~herzog/>.