presented at European Simulation and Modelling Conference, Le Havre, 2008

A Multiscale Simulation Environment for Performance Evaluation of high reliable heterogeneous Communication Networks

Andreas Lewandowski, Ralf Burda, Sebastian Subik and Christian Wietfeld
Communication Networks Institute (CNI)
Dortmund University of Technology, Germany
email: {andreas.lewandowski|ralf.burda|sebastian.subik|christian.wietfeld}@tu-dortmund.de

KEYWORDS

Multiscale Simulation Environment, Simulator Coupling, Performance Evaluation, Wireless Communication Networks

ABSTRACT

Mobile wireless communication networks can only be modelled by a bunch of parameter functions due to many uncorrelated influence factors. A Multiscale Network Simulation Environment (MNSE) for an exact effigy of all cross correlations is needed. Existing Cross-Layer approaches mostly respect only one single parameter system. The here described Multiscale Simulation Kernel based on OMNeT++ [1][2] considers the occasion scales Protocol, Radio Channel and User Mobility and provides a generic interface to add other simulation tools. Today's expansion stage combines the protocol simulator OMNeT++, a Radiowave Propagation Simulator, a 3D environment modeler and a Mobility Simulator. On account of an asynchronous coupling architecture, the distribution of simulation load over several simulation environments is possible. The implementation ensures a synchronous data exchange between single components.

INTRODUCTION

In order to generate a holistic model of a wireless communication network (WCN), the dynamic interaction between network infrastructure, environment and especially mobile objects must be considered. Mobile objects are not necessarily subscribers of the network as even a moving passive object may cause interaction with the radio channel.

The modeling of the wide range of effects of influence is mandatory for the design and operation of a wireless communication network [3]. The enhancing optimization calls for more detailed modelling, especially taking into account the interaction between system and environment.

Our Multiscale Network Simulation Environment (MNSE), described in this paper, focusses on the implementation of an abstract simulation kernel, where *best–in–class* simulation tools can be docked on by using standard TCP sockets. This architecture claims a quasi photo realistic model of the total system by taking into account interdependencies between the influencing variables.

Currently, **OMNeT++**, as a network simulator, is not able to model these dynamic influences accurately but is capable to be extended. In order to achieve the needed functionality, several claims are made by this modelling approach:

- A **3-D model** of the operating environment ensures the model of influences of the environment. The scenario model created by Google Sketch-Up is the base for the other simulation environments.
- A highly accurate **Ray Tracing Tool** is taken for the calculation of the radio channel properties. This Radio Channel Propagation Simulator (RPS) [4] is an industry standard tool and also used by mobile network operators.
- **Realistic model of user mobility** is created by using dedicated path and dynamic group mobility models. These mobility models are supported by the MObile Object Simulation Environment (MOOSE) [5] developed by CNI.
- Use of a numerical processing tool for **calculation and optimization** of PHY- and MAC-layer behaviour like bit error rate, adaptive modulation schemes and resulting data rates.
- The full model of dynamics is captured by the **Central Event Broker** (CEB) implemented in OMNeT++ which manages the major data exchange between the simulation environments. The simulation data is concentrated in a central data management system from where OMNeT++ can request relevant simulation results.

This paper proceeds by explaining the Multiscale Simulation Model assumption and the implemented simulation architecture in detail. A new technique for simulation setup is presented before we give a brief survey of the simulation flow and describe the used simulation coupling schemes. A detailed application scenario demonstrates the powerful compilation of MNSE before this paper is wrapped up by a short conclusion and an outlook to future improvements.

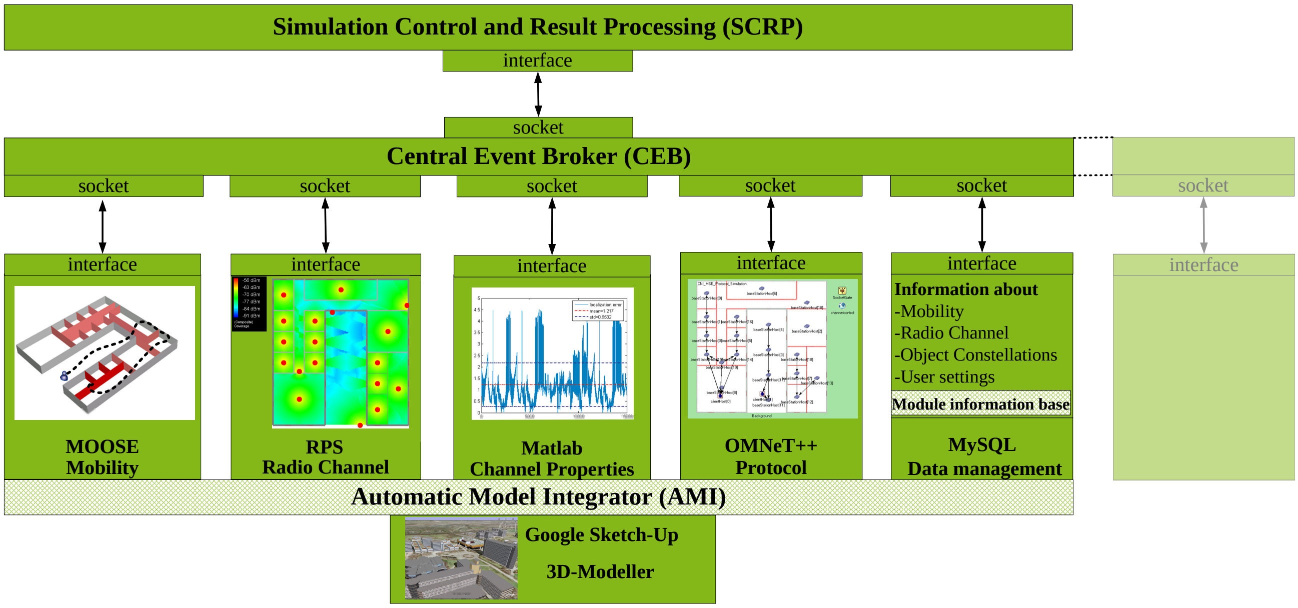

Figure 1: Implementation of the Multiscale simulation model

THE MULTISCALE SIMULATION MODEL

The Multiscale Simulation Model can be classified in **source models** and **system models** [6] which are able to interact by using the **Central Event Broker** (CEB). The source models generate so called *triggers*, some of which generate *messages* and also are the consumers of the latter. In short, any events which drive on a typical simulation are due to the source models.
A **source model** represents network participants which can be active or passive and delineates all kind of activity for each object. This includes generated network traffic, user mobility and transmission settings. Especially radio parameters like transmit power and antenna orientation are valid examples for a parameterized source model.
A **system model** is thought to represent the dynamic interaction between all active source models. At first, the physical models of the network participants, e.g. body model and received power have direct influence on the physical connectivity. These effects also result in a reverse impact on the source entities. Analog, mobility of active or passive objects can cause group effects which influence the mobility and communication behaviour of a single user.
A major challenge of MNSE is the efficient brokerage of events between sources and sinks as well as the coordination of different simulation tools. This task is taken over by the CEB in the overall model. The CEB ensures that events are reaching just the designated set of receivers. In a radio model this is needed in order not to hand over radio transmission events to nodes which would drop the signal anyway due to an insufficient signal level. Moreover, the parallel and yet synchronized execution of different simulation tasks distributed on a server cluster is a hot topic, yet not taken into account for this paper.

SIMULATION ARCHITECTURE

Figure 1 shows the simulation architecture where the central event broker is the core module which manages and interprets occurring in– and outputs. This approach ensures a synchronous behaviour of all subsidiary components. Synchronisation messages are created on demand, with which single threads can be stopped or started again. The whole system can be controlled with a **Simulation Control and Result Processing** (SCRP) console which ensures the scenario input and coherence between the different modelling bases of the simulation systems. A user interaction component is responsible for the GUI driven capturing of the application scenarios and an online result processing mechanism.
Single simulation systems can be accessed using standardized TCP sockets on which different *best–in–class* tools can be docked on.
Todays stage of expansion includes the mobility simulator MOOSE, a Radiowave Propagation Simulator (RPS), Matlab for numerical calculation of channel characteristics, OMNeT++ and the INET framework as shown in figure 1. Further important components of the MNSE are the individual interface adaptors between each tool and the CEB and the **Automated Model Integrator** (AMI), based on the **Model Information Base** (MIB), which provides information about interfaces and file formats. Google Sketch-up is used for the 3D environmental model of the scenario.

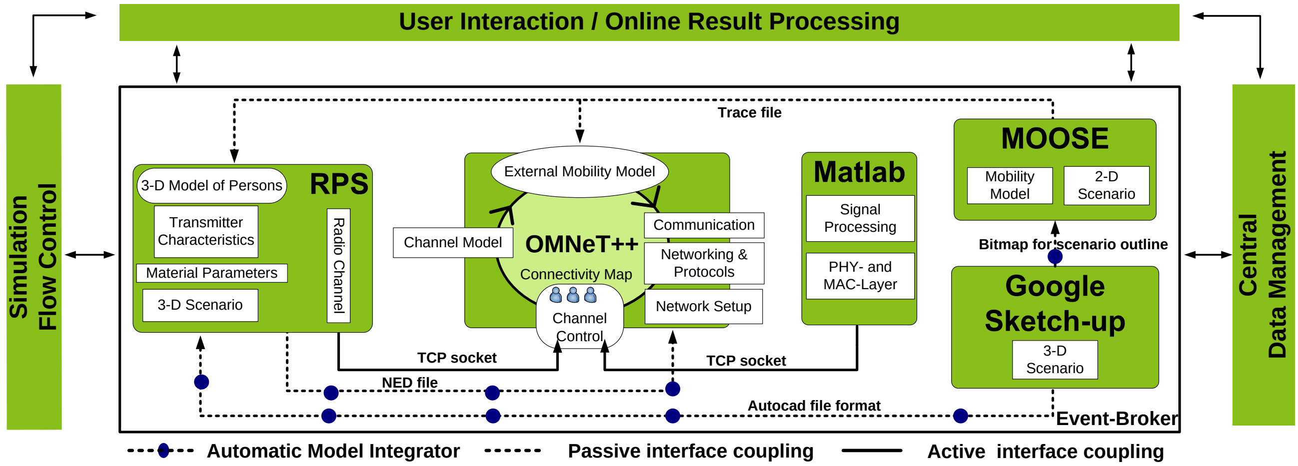

Figure 2: Detailed Interface Description

In fact the construction of interfaces between the CEB and simulation tools is a very demanding work. However, as being needed only in case of framework integration for a new tool, the effort is one time only and pays off rather fast during the deployment of MNSE.
Hence the open architecture ensures extensibility in terms of adding or exchanging simulation environments. Different types of interfaces (described later) are used for an optimal coupling of diversified simulation environments.

SIMULATION SETUP

The simulation setup is the critical task for fast engineering in this model approach. Several simulation and modelling environments need synchronous inputs for the initial setup. Especially coordinate transformations, transmitter settings, generation of mobility traces accordingly to the environment and the choice of appropriate OMNeT++ modules are very time consuming.
In order to minimize the input settings overhead, a so called Automatic Model Integrator (AMI) was developed which is able to mediate already done inputs for the outstanding simulation environments.
The process of simulation setup is now achieved in the following steps:

1. Generation of a 3D environment model in Google Sketch-up. This outsourcing from RPS 3D environment modeler is time saving because of a very intuitive user interface and Google Earth support.

2. The spatial model can then be imported in RPS via an AutoCAD file where material parameters are specified out of a huge material database. Transmitter settings like position, transmit power, operating frequency band and antenna configuration must be specified. A dedicated namespace for transmitter synonyms are chosen to be interpretable for AMI which is then able to specify appropriate network technologies and modules for OMNeT++ and MOOSE.

3. Now the AMI is able to create complete simulation setup files for the protocol and mobility simulation environments.

The setup process is accomplished.

COUPLING STRATEGIES FOR SIMULATION ENVIRONMENTS

Two different types of interfaces have been created, as not all coupled simulation environments require equal data exchange rates. A **passive** one (dashed lines in figure 2) and an **active** one (solid lines in figure 2) have been constructed and can be used and exchanged on demand.

The passive interface

The passive interface is considered to be a storage and retrieval process from a central database only. A generating simulator (e.g. MOOSE) will generate all information in a pre–run and store the results in the data base from which the CEB will retrieve a snapshot of all positions any time this is needed. This has been implemented by means of an MySQL data base and text trace files to achieve a high degree of flexibility.
The 3D modeler is also connected via a passive interface to RPS because the environment model is needed for radio propagation after construction in Google Sketch-up.

The active interface

The active interface is a superset of the passive one in the sense that the data source may update the information in the data base while the simulation is executing. As an example, coverage of a particular location in a wireless communication network may change due to fading effects if objects move. If the valid scope of information is just bilateral and temporal, a simple TCP socket connection is established between source and sink by the CEB. Furthermore, socket connections are used by the CEB to trigger update runs of the system model. A special performance issue is the connection to RPS. An integrated COM interface enables the use of RPS without the GUI. A given number of functions are accessible to change the properties of transmitters and the scenario by CEB.

SIMULATION FLOW

The following description demonstrates the dynamic interaction between simulation environments in a simulation flow graph as shown in figure 3. A 3D model of

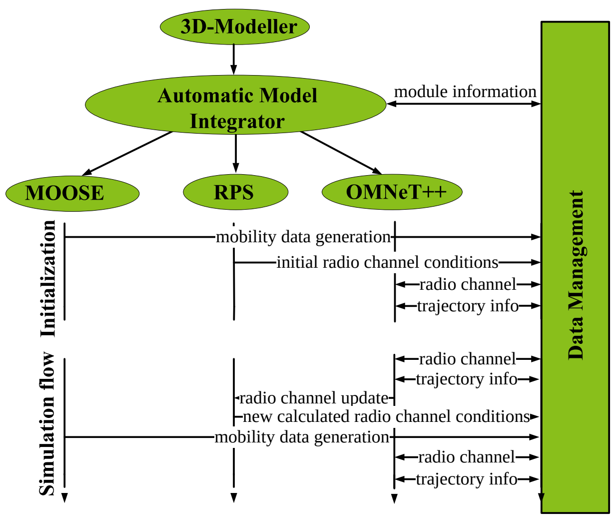

Figure 3: Simulation Flow Chart

the environment is generated by Google Sketch-up and transferred to Automatic Model Integrator where an initial scenario setup for RPS, MOOSE and OMNeT++ is generated. The source models have to be initialized afterwards because the position and the radio settings have influence on the initial coverage estimation.
That's why the initialization phase accomplishes calculation of mobility traces and radio channel data executed in parallel and stored in the data base, as MOOSE relies on a passive interface in this stage of expansion. In future releases it is planned that mobility can be influenced by the radio channel or the protocol e.g. if the received power is very low a person would go back to the position where the connectivity is better.
The protocol simulation can be started afterwards in order to grant access on relevant mobility and radio channel data.
The dynamic interaction between source and system model is viewable during the simulation execution. Whilst simulation, the influence of moving objects on the connectivity map is taken into account by a dynamic new calculation of the radio channel. The actual object constellation is assembled by the CEB and stored in the data base accessible for RPS. The integrated decision support function in the CEB decides whether the influence of moving objects on the radio channel is as big as a new calculation is meaningful.
The current object constellation is then taken by RPS for the new calculation. While RPS calculates the new radio coverage, OMNeT++ is able to proceed the simulation as far as possible. An integrated service thread handles the TCP connections to the external simulation environments.

EVOLUTIONARY PERFORMANCE EVALUATION TOWARDS FULL MULTISCALE FUNCTIONALITY

Time measurements of total simulation execution time are taken for a performance comparison between different expansion stages of the simulation environment. A simple WiFi scenario as shown in figure 4 was taken as reference and is characterized as follows:

- The number of nodes is increased incremental.
- Each node executes a simple *Pinging Application*.
- The scenario is formed by a simple $4 * 6$m room.
- A simulation time of 1 minute is examined.

Additional simulation environments have been added in four steps to reach the full multiscale functionality in order to show the impact of each tool on the performance of the whole simulation environment.

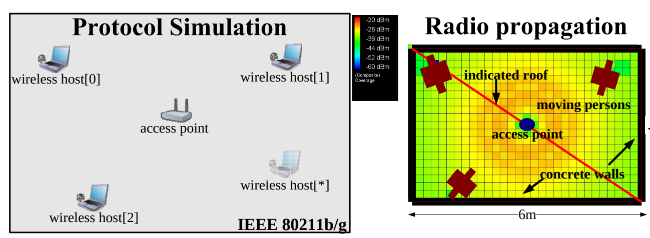

Figure 4: Simple WiFi Scenario for Performance Evaluation

1. The WiFi scenario is calculated in OMNeT++. Moving hosts are pinging the access point.
2. Mobility traces are calculated by MOOSE and queried out of a trace file.

3. A passive RPS interface is established where radio channel information is queried out of a MySQL database. The radio channel is not updated during the simulation.

4. Full multiscale functionality with an active RPS interface using intelligent decision support and point to point calculations for channel updates. Moving objects are updated in RPS with an object constellation file created by Central Event Broker.

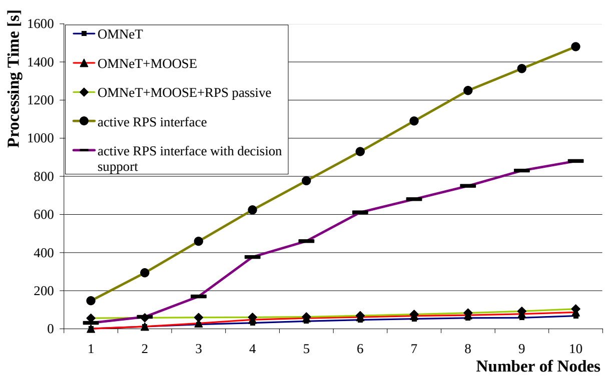

Figure 5: Simulation Time Analysis

The simple OMNeT++ simulation is the benchmark for performance comparison. As in Figure 5 depicted, each extension rises calculation time.
The integration of MOOSE mobility (*OMNeT+MOOSE*) needs additional queries of a mobility trace file and therefore the overall execution time rises unimposing.
It can be seen that the passive multiscale approach (*OMNeT+MOOSE+RPS passive*) takes more execution time because of database queries for radio channel data done here on every movement step. But the overall performance is still acceptable.
The active RPS interface (*OMNeT+MOOSE+RPS active*) is more accurate and does not rely on the passive radio channel assumptions taken for the previous interface. It shows a higher execution time because the radio channel is calculated online for all network participants during the simulation. Exact declarations about interference, bit error rate and adaptive retransmissions can be achieved by processing RPS results in Matlab. This high channel accuracy is especially needed in heterogenous network setups and indoor scenarios for e.g. evaluation of indoor localization protocols [9].
By applying a simple decision support function to the radio channel update process, higher processing efficiency can be achieved. For instance, not the whole scenario must be updated in case of a packet send event. Just local influences must be recalculated depending on objects' position change events.

APPLICATION SCENARIO: High reliable heterogenous communication network for rescue operations

In order to demonstrate the powerful compilation of MNSE, a modern heterogenous communication network for high reliable fire fighter communication is shown in the following chapter.
Todays fire brigades rely on traditional communication

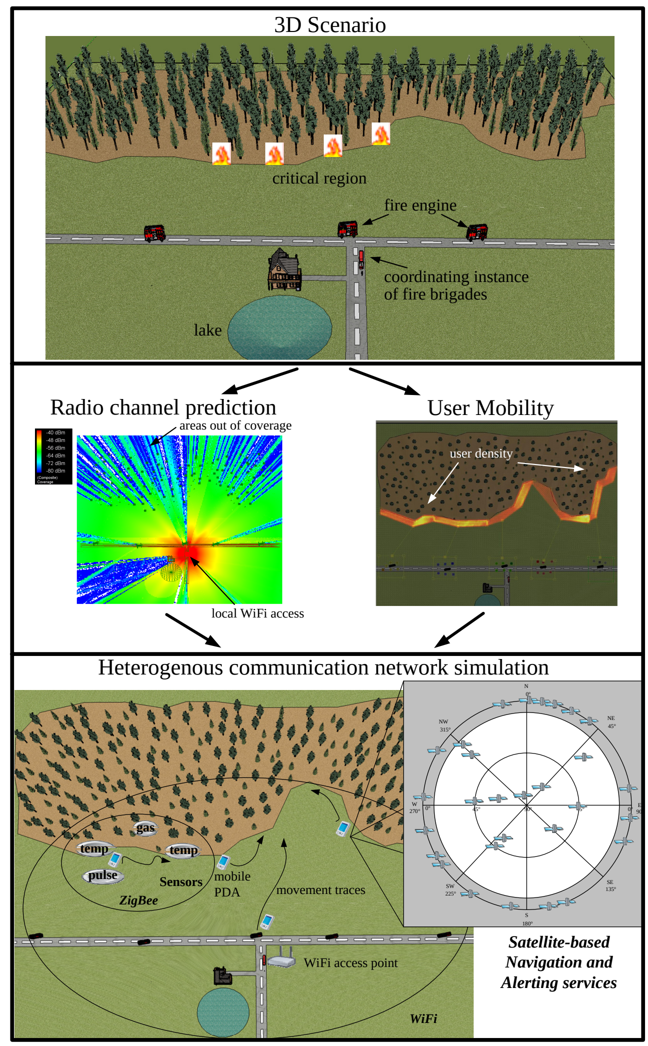

Figure 6: Forest fire scenario for evaluation of high reliable heterogenous communication services

and visualization techniques. In order to enhance the communication and efficiency in fire fighter missions, we are developing an innovative communication approach in the project Galileo4FireBrigades. The explanation of the concept is done by the simulation setup of a forest fire scenario shown in figure 6.
A 3D model gives detailed information of the place of action in this *scenario-based* approach. Fire engines arrive, fire fighters move out to the origin of fire, coordination instances receive information about occurring

events and the water supply must be granted.
For basic communication we propose to use group communication based on WiFi, as this technology operates independent on public telecommunication services like GSM or UMTS and can be used autartic. The radio propagation evaluation in figure 6 shows radio shadows e.g. in the forest, which can be balanced with so called mobile WiFi Dropped Units [8].
The mobility behaviour of the rescue forces seems to be intuitive. Highest densities are at the fire front as this process is a *line event*. This exact mobility model combining predefined areas and dedicated path mobility can be achieved by MOOSE.
The innovation of our proposed communication network is visible in the lower part of the scenario setup. Vital and environment sensors like pulse, temperature (environment and body), gas and position are transmitted over an IEEE802.15.4/ZigBee network to the group leader's PDA, which then forwards relevant sensor information to the coordination instance.
As we have already seen, in some cases, occurring radio shadows cannot be balanced by Dropped Units. The prospective enhanced Galileo Search–and–Rescue Service is then used for *communication out of coverage*. In [7] we have already shown using MNSE, that direct communication, especially in emergency situations, to the coordinating instance of the fire brigades is possible in near real time. Figure 6 shows the exact satellite constellation calculated by OMNeT++ relying on the Simplified General Perturbations [10] algorithm developed by NASA.

CONCLUSION AND OUTLOOK

This paper demonstrates the powerful compilation of the multiscale simulation approach by the given insight to architecture, simulation flow and performance aspects. Considering the dynamic interaction between scenario, radio channel and moving objects create a realistic model of a wireless communication network. The Central Event Broker architecture ensures a flexible and efficient simulation accomplishment and easy extension and exchange of best–in–class simulation tools.
Especially the performance evaluation of the full multiscale simulation environments leaves room for improvements. Intelligent decision support functions have to be developed to minimize the calculation overhead.
A real system demonstrator for the proposed communication infrastructure will be constructed out of the gained results from the use case scenario. We will then cross validate the real system with the multiscale simulation model.

ACKNOWLEDGEMENT

This work was funded by German Federal Ministry of Economics and Technology (BMWI).
Project: Galileo4FireBrigades – Employment of Galileo services for wide area fire fighter missions – 50NA0724

REFERENCES

[1] Andras Varga, The OMNeT++ Discrete Event Simulation System, Proceedings of the European Simulation Multiconference, 2001

[2] I Vari, INET Framework for OMNeT++, http://www.omnetpp.org/doc, last visited Aug. 2008

[3] Dhaou, R., Gauthier, V., Tiado, M., Becker, M., Beylot, A.: Cross Layer Simulation: Application to Performance Modelling of Networks composed of MANETs and Satellites, 2005

[4] J. Deissner, J. Huebner, D. Hunold, J. Voigt: RPS Radiowave Propagation Simulator, User Manual, www.actix.com

[5] S. Michaelis, C. Wietfeld: Comparison of user mobility pattern prediction algorithms to increase handover trigger accuracy, IEEE Vehicular Technology Conference, Melbourne, May 2006

[6] Kuehn, P. J.: Multiskalen Simulation – Versuch einer Systematik der Methoden und Ansaetze auf dem Gebiet der Verkehrssimulation, Workshop Simulation, ITG-FG 5.2.1, Bremen, Germany, 2006

[7] A. Lewandowski, B. Niehoefer, C. Wietfeld: Performance Evaluation of Satellite-based Search and Rescue Services: Galileo vs. Cospas-Sarsat, accepted for presentation on 68th IEEE Vehicular Technology Conference, Calgary, September 2008

[8] A. Wolff, S. Subik, C. Wietfeld: Performance analysis of highly available ad hoc Surveillance Networks Based on Dropped Units, 2008 IEEE Technologies for Homeland Security Conference, Boston, MA, USA, May 2008

[9] R. Burda, A. Lewandowski, C. Wietfeld: A Hybrid Indoor Localization Approach combining Meshing and Time of Arrival in IEEE 802.15.4 Networks, IEEE Vehicular Technology Conference, Singapore, May 2008

[10] T S Kelso, Felix Hoots, Ronald Roehrich, Spacetrack Report No.3 - Models for Propagation of NORAD Element Sets, December 1988