

Akteursbasierte Simulation in Virtuellen Welten

TORSTEN ASSELMEYER-MALUGA, PETER FRANK, ANDREAS HOHEISEL, HELGE ROSÉ
& BERTRAM WALTER

Institut FIRST, Fraunhofer Gesellschaft

Schlüsselwörter: Virtuelle Welt, Verteilte Umwelt-Simulationen, User Interaktion, Modell-Server

1. Einführung

Welche Folgen hat die Luftbelastung, die durch industrielle Produktion entsteht, für einen Bauern, der in der Nähe sein Feld bestellt? Wie wirken sich wiederum seine Aktionen auf die nahe gelegenen Wälder und Gewässer aus? Und wie kann man die Interessen verschiedenster Gruppen miteinander abstimmen, so dass gleichzeitig nachhaltige Entwicklungsstrategien entstehen. Wie können die Teilnehmer miteinander lernen, Problemlösungsstrategien entwickeln und verbessern?

Das von uns entwickelte M3(M3=Mensch-Modell-Messung)-Konzept, d.h. die Einbeziehung von Menschen in das Simulationsgeschehen, soll Antworten auf solche Fragen geben. Die M3-Technologie, d.h. die Vernetzung von Umweltmodellen durch die Repräsentanten in einer virtuellen Welt, ermöglicht einerseits die Implementierung einer Vielzahl von verschiedenen, untereinander gekoppelten Simulationsmodellen, welche jeweils bestimmte Bereiche der Umwelt und Wirtschaft simulieren. Beispielfähig implementiert sind das Wachstum von Pflanzen (Landwirtschaftsmodell, Forstmodell), Grund- und Oberflächenwasser (hydrologisches Modell), der Boden und dessen Einfluss auf das Pflanzenwachstum (Bodenmodell), Wetter und die Ausbreitung von Schadstoffen (Wetter- und Luftbelastungsmodell), Verkehr (Verkehrsmodell) und der Einfluss der Umwelt auf den Menschen (Impaktmodelle). Die Ökonomie soll endogen durch die Implementierung eines Wirtschaftssystems in das M3-System, d.h. in die Implementierung des M3-Konzeptes, abgebildet werden. Andererseits beinhaltet die M3-Technologie eine direkte Vernetzung der Teilnehmer durch eine interaktiven web-basierten Simulation. Dort bestimmen die Teilnehmer durch eigene Aktionen am PC die Abläufe und Entwicklungen in der virtuellen M3-Welt. Diese Aktionen haben natürlich auch Auswirkungen auf die virtuellen Repräsentanten der Teilnehmer selbst. So kann eine falsch aufgestellte Chemiefabrik zu einer Erkrankung führen.

Damit bieten sich Einsatzmöglichkeiten in der Untersuchung, Simulation, Erkenntnisvermittlung und im Management von komplexen Alltagssystemen in Wissenschaft, Wirtschaft, Politik, Bildung und Unterhaltung. In der Wissenschaft wird M3 als Integrationsinstrument eingesetzt, in der Wirtschaft als Ausbildungs- und Marketinginstrument. Der Bildung kann es zur Vermittlung komplexer Inhalte dienen, der Unterhaltungsindustrie als Basis einer neuen Generation realitätsbasierter Simulationsspiele.

2. Akteursbasierte Simulation

Die Simulation komplexer Alltagssituationen entwickelt sich derzeit zu einem entscheidenden Instrument der IT-unterstützten Problemlösung in Wissenschaft, Wirtschaft und Gesellschaft. Derzeit wächst das Interesse, interaktive Multiplayer Games in wissenschaftlichen Projekten anzuwenden. Beispiele dafür sind Multiplayer Business Games (Hardmann, van Rossum & van Bolhuis, 1995) und Internetforen für partizipatorisches Guppenlernen und Modellverbesserung (Hare, Gilbert, Medugno, Asakawa, Heeb & Pahl-Wostl, 2001). Diese Varianten setzen sowohl auf ein Mensch-Modell als auch auf Mensch-Mensch Interaktionen. Um dies zu realisieren, benötigt man eine Softwareinfrastruktur, die es ermöglicht ein Netzwerk von Umweltmodellen zu betreiben, die ein virtuelles Abbild der Umwelt realisieren, dass durch eine große Zahl agierender Teilnehmer verändert werden kann. Diese *Akteure* nehmen durch ihre Entscheidungen und Handlungen aktiv an der Simulation teil und können Problemlösungsstrategien entwickeln und testen. Andererseits erleben sie sofort die Konsequenzen ihrer Handlungen als Reaktion der Umweltsimulation und können so in einem evolutionären Lernprozess ihre Strategien verbessern. Dass dies keine Fiktion ist, zeigt eine kürzlich publizierte Untersuchung über Multiplayer On-linegames, die plausibel belegt, dass derartige Spiele wesentliche Aspekte des Verhalten aufweisen, wie es in realen ökonomischen System beobachtet werden kann (Castronova, 2001).

Als eine **Man-Model-Measurement (M3-)**Simulation (Rosé, 2001) oder auch M3-Simulation bezeichnen wir eine Simulation bestehend aus einem Netzwerk aus *Modellen*, die eine Welt simulieren, einer aktuellen Datenbank von *Messdaten* und einer großen Anzahl von realen *Menschen* als agierende und erlebende Teilnehmer.

3. Das M3-System

Das M3-System selbst besteht aus vier Hauptkomponenten: den Multi User Virtual Environment (MUVE), dem Modelserver, der Datenbank und dem Userinterface (Abb. 1).

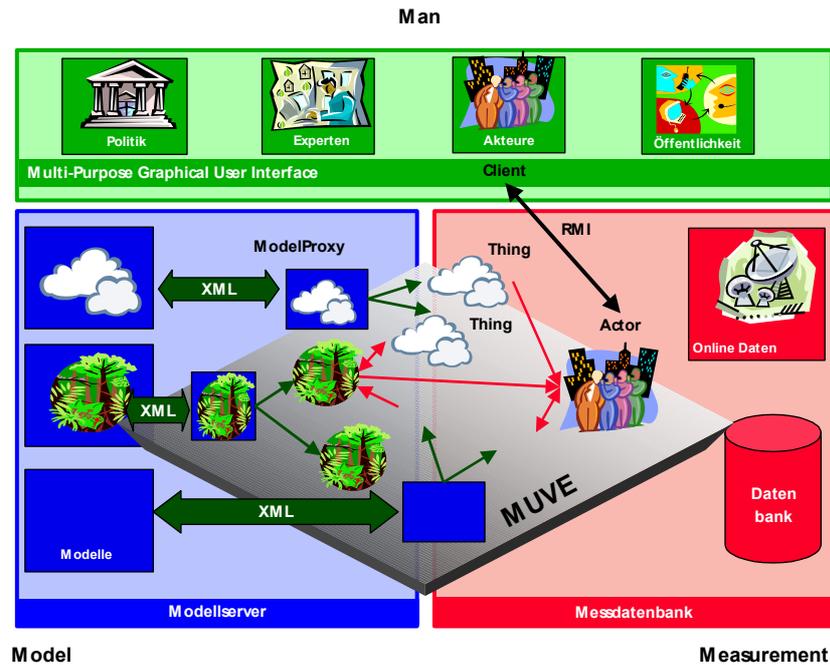


Abbildung 1: Struktur des M3-Systems

3.1 Multi User Virtual Environment

Das MUVE ist die logische Repräsentation des Wirklichkeitsbereiches, der untersucht werden soll und simuliert dessen Struktur durch ein Netzwerk von *Things* (Abb. 1). Die *Things* sind kommunizierende Softwarekomponenten, die die realen Objekte und ihre Wechselwirkungen simulieren. Ein entscheidender Vorteil ist die dadurch mögliche *objekt-orientierte* Modellierung des Realsystems. Jedes *Thing* besitzt hier einen *Zustand*, *Prozesse* der Zustandsänderung und *Relationen*, die seine Wechselwirkungen mit anderen Objekten beschreiben. Das in Java implementierte System realisiert eine Hierarchie von Basisklassen, die die wichtigsten Objekteigenschaften, wie z.B. relational, veränderlich, lokalisiert, sichtbar, beweglich, mechanisch, etc., zur Verfügung stellt. Von diesen elementaren *Things* können dann leicht konkrete Dinge, wie Kornfelder, Häuser, Autos, Pflanzen, etc., abgeleitet werden. Dabei starten wir mit der einfachsten Klasse *Thing*, die durch Implementierung von Interfaces und durch Ableitungen erweitert. Dieses Konzept (Implementierung eines Interfaces und dann Ableitung) ermöglicht es, das Problem der Uneindeutigkeit von Mehrfachableitungen zu umgehen. Die Klasse *Actor* repräsentiert den Spielteilnehmer. Sie stellt die mechanischen und biologischen Eigenschaften, wie z.B. Energieverbrauch, Nährstoffhaushalt, Gesundheitsindex, zur Verfügung. Außerdem implementiert sie die Netzwerkfunktionalität, die die Verbindung zwischen der Actor-Instanz im MUVE und des 3D-Userinterface des Teilnehmers am heimischen PC realisiert. Der derzeitige Prototyp des MUVE simuliert ein vier Quadratkilometer großes Gebiet um den Berliner Alexanderplatz mit einigen charakteristischen Gebäuden, Straßen und computergenerierten Autos die Abgase produzieren.

Externe Umweltmodelle werden in das MUVE durch spezialisierte *Things* integriert. Diese *ModelProxies* werden dynamisch instanziiert und mit den korrespondierenden Zustandsvariablen der *Things* verknüpft. Jedes externe Modell wird durch seinen

eigenen ModelProxy repräsentiert, der die Modelldaten an die Things weiterleitet. Durch dieses Konzept werden die Modelle untereinander durch eine Kommunikationsstruktur verknüpft, die sich aus den Relationen der Things ableitet. Modell und Proxy kommunizieren dabei durch ein XML-basiertes Protokoll (siehe <http://mmm.first.fhg.de/papers/modellkopplung.html> oder http://mmm.first.fhg.de/papers/hoheisel_m3_iemss2002.pdf), dass in unserem Falle eine hohe Flexibilität und leichte Erweiterbarkeit garantiert.

Während der Laufzeit kann durch eine Trennung der Verbindungen des alten ModelProxy und das Knüpfen der Verbindung zu einem anderen Proxy das neue Modell eingefügt werden. Dies ermöglicht die schnelle Evaluation und den Test alternativer Modellimplementierungen ohne Neustart des Gesamtsystems oder Softwareanpassung der Things. Dabei unterstützt der ModellServer den Transfer der internen Zustände zwischen den Modellen und ermöglicht die leichte Integration in das M3-System.

3.2 Der Modellserver

Der Modellserver des M3-Systems realisiert ein Netzwerk von Umweltmodellen, die durch generische Daten- und Steuerschnittstellen verbunden sind. Das allgemeine Problem der Integration heterogener Implementierungen wird damit angegangen. Verfügbare Simulationsmodelle liegen meist in unterschiedlichen Programmiersprachen wie FORTRAN, C oder C++ vor und verwenden diverse Datenformate. Viele dieser Modelle werden als Stand-Alone Anwendungen auf unterschiedlichen Hardwareplattformen genutzt. Bei der Entwicklung dieser Modelle wurde auch meist nicht die Kopplung mit anderen Modellen vorgesehen. Im M3-System nutzen wir daher das Konzept von *Wrappern*, die die externen Simulationsmodelle kapseln und Standardschnittstellen zur Verfügung stellen. Im Gegensatz dazu gibt es noch *ModellProxy*, die der Repräsentant des Modells in der virtuellen Welt sind. Als Beispiel betrachten wir das Landwirtschaftsmodell SWIM. Dieses Simulationsmodell ist in FORTRAN geschrieben und von einem entsprechenden Wrapper umgeben, der sowohl die Programmsteuerung als auch den Datenaustausch mit dem M3-System regelt. Der direkte Repräsentant von SWIM in der virtuellen Welt ist ein Feld, welches im Kontakt zum ModellProxy, als abstrakten Repräsentant, steht. Die Entscheidung fiel dabei auf XML (Extensible Markup Language) als Datenaustauschsprache.

Die in vielen Fällen verwendete enge Kopplung von Modellen z.B. durch Shared Memory erfordert im Allgemeinen einen hohen Integrationsaufwand und erhebliche Änderungen an den verwendeten Modellimplementierungen. Im M3-System werden daher die Modelle durch die Verwendung asynchroner, d.h. auf verschiedenen Zeitskalen stattfindender, Kommunikationen nur lose gekoppelt. Diese Kopplung ist netzwerkorientiert, sehr flexibel und ermöglicht die einfache Wiederverwendung von Modellen in unterschiedlichen Kontexten. Außerdem erfolgt die Ankopplung dynamisch in der Laufzeit, so dass keine starren Kopplungsstrukturen von der Compilierung festgelegt werden müssen. Dies ermöglicht es uns, Modelle während der Simulation auszutauschen ohne die Simulationswelt neu starten zu müssen.

Die Steuerung dieses Netzwerkes aus Simulationsmodellen ist im M3-System durch RMI (Remote Method Invocation) realisiert. Die Modellwrapper stellen einen entsprechenden RMI-Service zur Verfügung, der über das Netzwerk die Modelle starten, stoppen und steuern kann. Dabei werden unter Umständen auch für das Starten

benötigte Informationen mitgeliefert. Wird ein Modell gestoppt, so behalten die noch laufenden Modelle die letzten Werte bis neue geliefert werden können. Die Wrapper nutzen dabei modellseitig die normale Standardein- und -ausgabe des externen Simulationsprogramms zur Kommunikation. Dies ist ein entscheidender Vorteil da jede Programmiersprache eine solche Schnittstelle zur Verfügung stellt.

Die Testversion des M3-Systems implementiert bereits folgende Modelle:

SWIM (Soil and Water Integrated Model) simuliert den Wasserhaushalt, die Vegetation, Erosion und die Nährstoffdynamik ganzer Regionen. Es wurde am Potsdam Institut für Klimafolgenforschung (PIK) entwickelt und in FORTRAN implementiert (Krysanova, Roosaare, Meiner & Vasilyev, 1989).

4C simuliert das Wachstum von Waldbeständen einschließlich der Hydrologie und wurde ebenfalls am PIK entwickelt und in FORTRAN implementiert (Bugmann, Grote, Lasch, Lindner & Suckow, 1997).

WGEN ist ein Wettergenerator von Richardson und Wright. Auf der Basis von statistischen Daten produziert dieses Programm charakteristisches Wetter für eine Region (Richardson & Wright, 1984). Implementierung: FORTRAN.

REWIMET ist ein hydrostatisches drei Schichten Luftschadstoffmodell. Es wurde mit einem Lagrangian Modell zur Simulation von PM10 Rußpartikeln gekoppelt und wird bei FIRST betrieben (Unger, Gerharz, Mieth & Wottrich, 1998). Implementierung: C.

Impact ist ein Modell zur Simulation des Einflusses der Umweltschadstoffe auf die Gesundheit (Beger, Bieninda & Gester, 2001). Implementation: Java.

Traffic ist ein mesoskopisches Verkehrsmodell und wurde in Java implementiert (Schmidt, Schäfer & Nökel, 1998).

3.3 Der Datenbankserver

Der Datenbankserver verwaltet alle Informationen zum Start und Konfiguration der Modelle, der Things und des MUVE. Gleichzeitig protokolliert er alle relevanten Informationen der Simulationsläufe. Dieser netzbasierte Datenbankservice wird durch MySQL realisiert und kann durch eine JDBC-Schnittstelle vom MUVE angesprochen werden. Außerdem ist ein Zugriff und die Verwaltung der Datenbank durch einen WEB-Browser möglich.

3.4 Das 3D-Userinterface

Die von MUVE und Modellserver generierte, virtuelle Welt wird von den Akteuren bevölkert. Dazu nutzten die Teilnehmer ein spezielles 3D-Userinterface, welches ebenfalls in Java geschrieben ist. Für jede Nutzergruppe (Experten, Akteure, Entscheidungsträger, etc.) kann das Interface unterschiedliche *Views* der simulierten Welt liefern, die die für diese Nutzergruppe relevanten Informationen darstellen. Die Implementierung des 3D-Interface' der Akteure ist bereits weit fortgeschritten und stellt die simulierte Welt in einer intuitiven dreidimensionalen Form dar (Abb. 2). Eine genaue Beschreibung der Bedienung etc. findet sich unter <http://mmm.first.fhg.de>.

Zur Visualisierung wird die von uns erweiterte Open Source 3D-Engine Opale Soya verwendet, die ein hardwaregestütztes Rendering über die OpenGL-Schnittstelle erlaubt. Zur Nutzung vorhandener 3D-Objekte wurde ein 3ds-Converter implementiert, der das weitverbreitete 3D Studio Max unterstützt. Die Figuren der Akteure wurden in Poser erstellt. Die anderen Komponenten des 3D-Interface wurden in Java Swing implementiert, wobei das Listener-Konzept (Observer-Pattern) zum Einsatz kam. Die Kommunikation mit dem MUVE erfolgt über RMI-Verbindungen. Damit ruft der Akteur durch eine Aktion eine Methode im MUVE auf, die beim nächsten Zeitschritt ausgeführt wird.

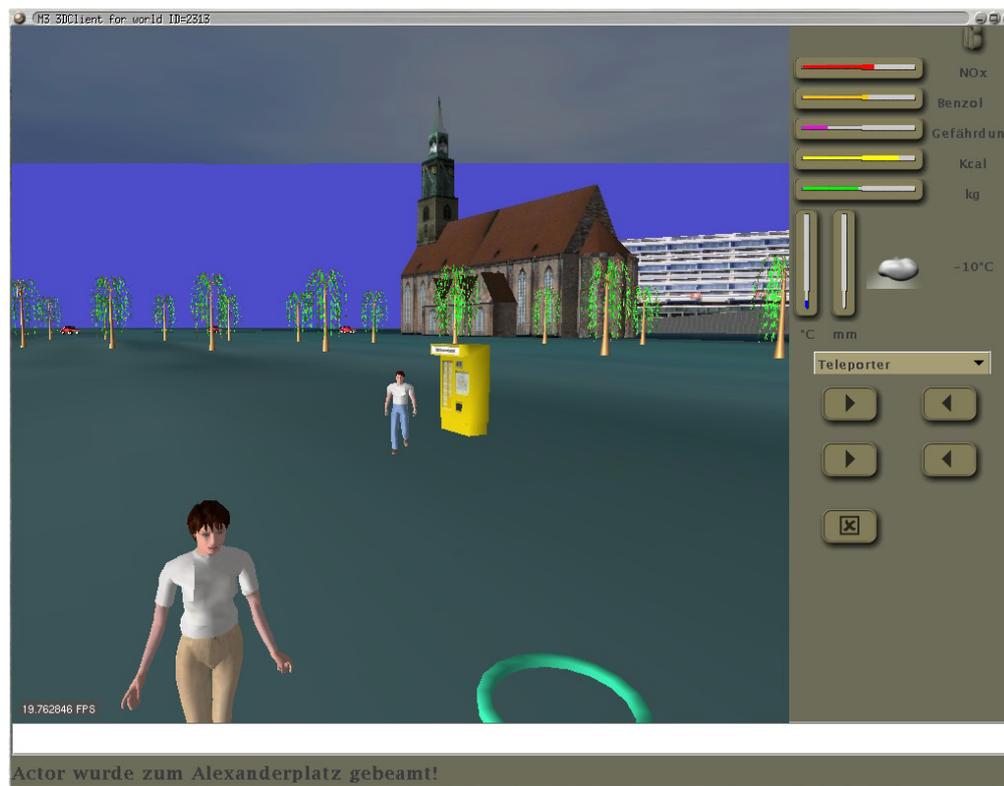


Abbildung 2: Das 3D-Userinterface für die Akteure

Um eine einheitliche Umgebung für alle simultanen Teilnehmer zu realisieren, werden die Positionen aller beweglichen Objekte (Akteure, Autos) im MUVE zentral berechnet. Die neuen Positionen berücksichtigen dabei Kollisionen und Umgebungseinflüsse (z.B. Reibung, Schwerkraft) und werden vom MUVE an die 3D-Interfaces der Akteure gesendet, die dann alle sichtbaren Objekte darstellen.

4. Ausblick

Das M3-Projekt ist ein sehr anspruchsvolles Unterfangen. Zweifellos kann der hohe Aufwand, der zu seiner Verwirklichung notwendig ist, nicht von einzelnen Teams von Informatikern, Umweltwissenschaftlern oder Modellierern erbracht werden. Vielmehr ist eine weitreichende Kooperation aller dieser Entwicklungsgruppen eine unabdingbare Voraussetzung für die erfolgreiche Umsetzung der Vision. Das M3-System selbst, bietet dafür eine geeignete Kommunikationsplattform. Aufgrund seines interaktiven Charakters kann es als seine eigene Entwicklungsumgebung dienen, die eine schnelle Kommunikation und Kooperation sichert. Das M3-System ist nicht

bloß eine „Software“ – es ist eine simulierte Welt, bevölkert von realen Menschen und hat daher die Fähigkeit sich selbst weiter zu entwickeln. Diese Art der *evolutionären Softwareentwicklung* macht den Erfolg von Open Software Projekten wie Linux oder GNU aus (Raymond, 2001).

Alle Nutzer können durch neue Implementierungen zur Entwicklung des Systems beitragen. Zur Weiterentwicklung des Systems werden mehrere „Testwelten“ gestartet, die jeder durch eigene Beiträge beliebig verändern kann. Änderungen, die sich hier bewähren und von einer Jury als funktional, stabil und nützlich bewertet werden, werden dann in die offizielle M3-Welt übernommen. Diese Art der Entwicklung ist in der Tat ein evolutionärer Prozess und eine Studie an der Harvard Business School belegt, dass diese Art der Softwareentwicklung die größten Erfolge verspricht (McCormack, 2001).

Der derzeitige Prototyp des M3-Systems zeigt die prinzipielle Machbarkeit des M3-Konzeptes. Solche Simulationssysteme können dabei durchaus in der Lage sein, realistische Aspekte von wirtschaftlichen Gegebenheiten nachzubilden. Es bleibt allerdings zu untersuchen, ob auch komplexe sozio-ökonomische Systeme hierdurch beschreibbar werden und wie dabei das diffizile Problem der Kopplung verschiedener Modelle befriedigend zu lösen ist. In jedem Fall stellt der Einsatz des M3-Systems in einer wohldefinierten und limitierten Problemsituation ein neues und vielversprechendes Instrument im Verstehen, Lernen und Lösen all jener komplexen Situationen dar, die unser alltägliches Leben für uns bereit hält.

5. Literatur

Beger, E.; Bieninda, D. & Gester, S. (2001). *Modul Impactmodelle im M3-System des Projektes GLOBALSIM*. Interner Technischer Bericht, IBB Ingenieurbüro Beger, 2001.

Bugmann, H.; Grote, R.; Lasch, P.; Lindner, M. & Suckow, F. (1997). A new forest gap model to study the effects of environmental change on forest structure and functioning. In G.M.J. Mohren and K. Kramer, editors, *Global Change Impacts on Tree Physiology and Forest Ecosystems*, pages 255--261. Kluwer Academic Publishers, 1997.

Castronova, E. (2001). *Virtual worlds: A first-hand account of market and society on the cyberian frontier*. CESifo working paper No. 618, submitted to netnomics, 2001.

Hardman, L.; van Rossum, G. & van Bolhuis, A. (1995). *An interactive multimedia business game*. *Journal of Intelligent Systems*, 5, 151--177, 1995.

Hare, M.; Gilbert, N.; Medugno, D.; Asakawa, T.; Heeb, J. & Pahl-Wostl, C. (2001). The development of an internet forum for long-term participatory group learning about problems and solutions to sustainable urban water supply management. In L.M. Hilty and P.W. Gilgen, editors, *Sustainability in the Information Society*, pages 743--750, Marburg, Metropolis Verlag, 2001.

Krysanova, V.; Roosaare, J.; Meiner, A. & Vasilyev, A. (1989). *Simulation modeling of the coastal waters pollution from agricultural watershed*. *Ecological Modeling*, 49, 7--29, 1989.

MacCormack, A. (2001). *Product-development practices that work: How internet companies build software*. MIT Sloan Management Review, 42, 75--84, 2001.

Raymond, E.S. (2001). *The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*. OReilly & Associates, 2001.

Richardson, C.W. & Wright, D.A. (1984). *Wgen: A model for generating daily weather variables*. Technical report, U. S. Department of Agriculture, Agricultural Research Service, 1984.

Rosé, H. (2001). M3-Simulation - Multidimensional Modeling of Sustainability Strategies in Virtual Worlds. In K. Bellmann and C. Landauer, editors, *Proceedings of the International Conference on Virtual Worlds and Simulation*, Phoenix, Arizona, 2001.

Schmidt, M.; Schäfer, R.-P. & Nökel, K. (1998). SIMTRAP: *Simulation of traffic-induced air pollution*. TRANSACTIONS of SCS, 15, 122--132, 1998.

Unger, S.; Gerharz, I.; Mieth, P. & Wottrich, S. (1998). *HITERM - high-performance computing for technological risk management*. TRANSACTIONS of SCS, 15, 109--114, 1998.