

# KALIBRIERUNG VON MULTIAGENTENMODELLEN

Christoph Oechslein, Franziska Klügl und Frank Puppe

Lehrstuhl für Künstliche Intelligenz,  
Universität Würzburg  
Am Hubland, 97074 Würzburg

{oechslein,kluegl,puppe}@informatik.uni-wuerzburg.de

## 1 Motivation

Mit der Steigerung der verfügbaren Rechenpower wurden auch neue Formen der individuenbasierten Simulation möglich. Eine der mächtigsten und flexibelsten ist die Multiagentensimulation [Dor96]. Obwohl die Konzeptionalisierung und Modellierung eines biologischen oder sozialen Systems auf der Basis von Agenten oftmals einfacher ist als eine Makromodellierung [Jud94], ist die Implementierung und Validierung eines derartigen Modells wegen der inhärenten Parallelität sehr komplex.

Wie bei anderen Formen der Modellierung läßt sich auch die Implementierung eines Multiagentenmodells in zwei Phasen unterteilen: Eine Phase, in der die drei Bestandteile eines Multiagentenmodells, das Verhalten der einzelnen Agenten, die Interaktionen zwischen Agenten und das Verhalten der Umwelt (qualitativ) festgelegt werden. Diese Phase kann durch eine adäquate Modellierungsumgebung mit vorgegebener Agentenarchitektur weitgehend unterstützt werden [Dec96, KP98]. In der zweiten Phase der Modellkonstruktion soll das Verhaltensmodell so kalibriert werden, daß das in der Simulation generierte Verhalten möglichst dem im entsprechenden Realitätsausschnitt beobachtbarem entspricht.

Diese zweite Phase der Parameteranpassung ist bei Multiagentenmodellen besonders problematisch, da die Parameter das rein lokale Verhalten der Agenten und ihrer Interaktionen bestimmen, während das Zielverhalten für das Gesamtsystem, also auf der Makroebene, definiert ist. Die variierbaren Parameter können durch den Modellbauer, möglicherweise auch abhängig von der vorgegebenen Agentenstruktur (halb-)automatisch, identifiziert werden. Ebenso leicht zugänglich ist die explizite Definition des Zielverhaltens, bzw. der Makrostrukturen. Allerdings sind die genauen Auswirkungen der Parameter auf das Zielverhalten, insbesondere in komplexen Modellen schwer zu überschauen und zu kalibrieren.

Grundsätzlich können auch bei Multiagentenmodellen Standard-Optimierungsverfahren verwendet werden. Problematisch ist dabei allerdings, daß Simulationsexperimente mit vollständigen Multiagentenmodellen sehr aufwendig sind und so Verfahren, bei denen eine Vielzahl von Bewertungen einzelner Parameterkombinationen notwendig sind, nicht praktikabel scheinen. Besser wäre dagegen ein Vorgehen, bei dem das Multiagentenmodell in ein aggregiertes und somit vereinfachtes Modell überführt würde, das dennoch den Wirkungszusammenhang der relevanten Parameter widerspiegelt. Experimente mit diesem Modell könnten Hinweise geben, welche Parameterwerte eine Lösung des oben skizzierten Kalibrierungsproblems darstellen bzw. zumindest Constraints für die Lösung oder den Lösungsweg aufzeigen. Allerdings ist die

Konstruktion eines derartigen Modells alles andere als einfach, und möglicherweise sogar nicht automatisch machbar [Dar94].

In dem nächsten Abschnitt werden wir zunächst allgemein auf Kalibrierungsmöglichkeiten eingehen, die in Multiagentenmodellen möglich sind. Im Anschluß daran wird versucht, diese dann auf ein konkretes Szenario, das „Waldbrandszenario“ anzuwenden. Den Schluß bildet ein Abschnitt über unsere Erkenntnis aus obigen Schritten und die geplanten Erweiterungen unseres eigenen Simulationstools SESAM[KP98].

## **2 Unterstützung der Kalibrierung**

### **2.1 Manuelle Kalibrierung**

Eine ‚normale‘ Kalibrierung wird meist mittels eines ‚Trial&Error‘ Verfahrens durchgeführt: Während oder nach einer Simulation werden verschiedene Parameter vom Modellbauer angepaßt. Der somit veränderte Simulationslauf wird weiter beobachtet und dient als Grundlage für weitere Parameteradaptionen. Hier steht also eine Kalibrierung auf phänomenologischer Ebene im Vordergrund, wobei der Modellbauer meist ein globales Verhalten innerhalb gewisser Intervalle oder Gleichgewichtszustände, also eine *Zielrelation* erreichen möchte.

Falls Simulationsläufe absolut oder relativ zueinander bewertet werden können, eignen sich Standardsuchverfahren wie z.B. Hill-Climbing. Besonders attraktiv sind hierbei stochastische Verfahren, wie z.B. Genetische Algorithmen oder Simulated Annealing, da sie große Suchräume effizient durchsuchen können [Gol89]. Nachteil bei diesen Verfahren ist der hohe Zeitbedarf, der meist für nur einen einzigen Simulationslauf benötigt wird. Da diese Verfahren sehr viele Läufe ausführen müssen, um gute Einstellungen zu finden, sind sie nur bei sehr kleinen oder abstrakten Modellen einsetzbar.

### **2.2 Kalibrierung mit Hilfe von Modellaggregation und -abstraktion**

Eine weitere Möglichkeit zur Modellkalibrierung basiert auf der Konvertierung des Mikromodells zu einem Makromodell. Ein Makromodell kann durch (Un-)Gleichungssysteme oder durch abstrakte Verhaltensmodelle beschrieben werden. Mit Hilfe dieses aggregierten und abstrakten Modells können nun abhängig von der Struktur des Modells verschiedene Kalibrierungsmethoden angewandt werden. Diese beinhalten zum einen die im vorherigen Abschnitt 2.1 erwähnten Möglichkeiten zur Anpassung von Parametern bei Simulationsläufen, was bei Verhaltensmodellen die zu präferierende Lösung ist. Die andere Möglichkeit besteht im Einsatz von Methoden aus der Numerik zum Lösen von Gleichungssystemen oder von Optimierungsproblemen.

Problematisch bei diesem Vorgehen ist dabei die Erstellung des aggregierten und abstrakten Modells. Eine vollautomatische Erstellung ist kaum vorstellbar, sie würde einer automatischen Umwandlung eines Mikromodells mit lokalen Interaktionen in ein Makromodell gleichkommen, was angesichts von Phänomenen wie Emergenz ein (zu?) ehrgeiziges Ziel ist. Ein anderer Ansatz würde die zugrundeliegende Mikromodellstruktur ignorieren und auf der Basis von Simulationsdaten versuchen, das Makroverhalten als Blackbox-Verhalten zu lernen. Dies wäre eine Anwendung des ‚Lernen durch Vormachen‘-Paradigma [SZ95]. Die Problematik hierbei bestünde jedoch in der Abbildung der zu kalibrierenden Mikromodellparametern zurück aus den gelernten Makromodellparametern.

In der Literatur finden sich hierzu verschiedene Ansätze für Modellierung auf mehreren Ebenen. ‚Multi-Modelle‘ werden vor allem bei der Simulation physikalischer oder technischer Systeme [FZ92,Fis95] eingesetzt. Dabei wird das Verhalten eines Systems zustandsabhängig mit unterschiedlich verfeinerten Modellen beschrieben, was einen großen Modellierungsaufwand zur Folge hat. Mehrebenenmodelle findet man vor allem in sozialwissenschaftlichen Simulationen [Tro96,Möh96], deren mikroanalytische Modelle keinesfalls die Komplexität an Interaktionen betrachten, die in Multiagentenmodellen möglich sind. Es gibt neuere Ansätze, bei denen Multiagentensysteme mit zusätzlichen Strukturen auf der Gesellschaftsebene beschrieben werden, z.B. [MG98], die allerdings sehr einfache Agenten auf der unteren Ebene verwenden. Die Gesellschaftsebene ist dabei vor allem für die Ausgabe bei den Simulationsläufen verantwortlich. Unsere konkreten Vorstellungen, wie man Mechanismen zur Unterstützung der Kalibrierung in ein Simulationstool einbauen kann, wird erst in Kapitel 4.3 angegangen, zunächst wollen wir kurz die vorhandenen Modellierungsmöglichkeiten in SESAM vorstellen und dann anhand eines konkreten Szenarios verdeutlichen, wie ein Mensch mit ‚Papier und Bleistift‘ bei der Kalibrierung vorgehen würden.

### 3 Modellierung in SESAM

Das Werkzeug SESAM (Shell for Simulated Agent Systems [KP98]) bietet eine generische Umgebung zum Modellieren und Experimentieren mit agentenbasierten Systemen. Mit ihm sollen Fachexperten in die Lage versetzt werden, ohne Kenntnis einer Programmiersprache komplexe Modelle einfach zu erstellen und zu testen. SESAM versucht so wenig wie möglich die Generalität des zu erzeugenden Modells aufgrund der einfachen Erstellbarkeit einzuschränken. In SESAM kann die Umgebung durch zweidimensionale diskrete oder kontinuierliche Karten modelliert werden. Eine ‚dritte‘ Dimension kann durch Verknüpfungen zwischen verschiedenen Karten erreicht werden. Jedes Feld kann beliebig viele Objekte tragen, ein Objekt kann dabei entweder ein Agent, der sich und seine Umgebung verändert oder eine Ressource sein, die die Umwelt durch ihre Existenz bereichert. Eine Erweiterung ist die Hinzunahme von Umgebungsereignissen, die Änderungen unabhängig von den Agenten bewirken können.

Ein Agent wird in SESAM durch drei Komponenten spezifiziert:

- Sensorische Aktionen und interne Parameter,
- seine Aktionsselektionsstrategie und
- seine effektorischen Aktionen.

Hierbei kann der Modellbauer auf einen großen Fundus von domainunabhängigen Standardprimitiven zugreifen, um so seine domainabhängigen Aktionen grafisch einzugeben. Diese Aktionen führt der Agent in Aktivitäten aus, welche dabei entweder aus primitiven Aktionen oder wieder aus komplexen Aktivitäten mit UND-, ODER-Verknüpfungen bestehen können. Diese Aktivitäten werden dann durch Regeln zu einem Graph verbunden, wobei sich der Agent immer in *einer* dieser Aktivitäten befindet. Komplexe Aktivitäten können auch als UND/ODER-Graph (Skelettplan) eingegeben werden; diese Darstellung ermöglicht eine einfache und übersichtliche Eingabe für viele Modelle. Um in bestimmten Situationen schnell auf Änderungen der Umgebung reagieren zu können, existieren sog. Notregeln, die in jeder beliebigen aktuellen Aktivität des Agenten geprüft werden, diese dann unterbrechen und andere Aktivitäten starten können. Eine weitere Modellierungshilfe stellen Verhaltensstereotypen dar, mit denen die Sammlung

von Verhalten, Entscheidungsplänen und Notregeln einfach strukturiert werden können. Damit ist eine transparente Darstellung eines Modells möglich.

In dieser Simulationsumgebung wurden neben biologischen Modellen [KPRT98, DKPT98] u.a., auch das Waldbrandszenario erstellt. Auf dieses geht das folgende Kapitel genauer ein.

## 4 Beispiel: Kalibrierung im Waldbrandszenario

### 4.1 Beschreibung

Das Waldbrand-Szenario (siehe Abbildung 1) wird am Lehrstuhl für Psychologie II in Bamberg benutzt, um menschlichen Vorgehensweisen in komplexen Problemlöse-Szenarios zu untersuchen [DS92]. Dabei werden Versuchspersonen vor die Aufgabe gestellt, Waldbrände zu löschen, indem sie mehrere Feuerlöschleinheiten steuern. Abhängig von allgemeiner Temperatur entstehen mehrere Brandherde, die sich abhängig von Windrichtung, Windstärke, u.a. ausbreiten und Wald, Siedlungen und Stauwehre zerstören. An Feuerlöschleinheiten stehen der Versuchsperson zum einen Hubschrauber, die relativ schnell über das Spielfeld fliegen, aber nur mit wenig Wasser löschen können zur Verfügung. Zum anderen existieren Raupen, die zwar langsam fahren, dafür aber neben längerem Löschen zusätzlich Schneisen schlagen können. Der Wasservorrat der Einheiten kann in den Stauseen wieder aufgefrischt werden.

Interessant ist in diesem Szenario, welche Strategien die Versuchspersonen verfolgen. Wo positionieren sie die Feuerlöschleinheiten relativ zum Feuer, wann und wo lassen sie die Raupen roden? Verändern sich die strategischen und taktischen Entscheidungen, wenn die Versuchsperson zunehmend unter Druck gerät, weil mehr und mehr Wald brennt.

### 4.2 Kalibrierung

Zunächst stellt sich die Frage, was in diesem Szenario zu kalibrieren ist, denn anders als z.B. in einem biologischen Modell mit vielen Parametern, gibt es kein in der Natur beobachtbares Makroverhalten, an das das Modellsystem angeglichen werden sollte. In diesem Szenario (und auch in anderen Spielszenarios) besteht eine Kalibrierung darin, die Umgebung so auszurichten, damit der Spieler überhaupt eine „reelle Gewinnchance“ hat. Dies ist im besonders für die Motivation der Probanden wichtig.

Kalibrierung bedeutet also im Waldbrandszenario, daß die Einstellung der Parameter ‚fair‘ sind. Da der Begriff der Fairness sehr subjektiv ist, haben wir uns auf Grenzwerte konzentriert, also: Kann der Spieler in diesem Szenario überhaupt gewinnen? Unsere Herangehensweise war durch eine Analyse des Szenarios, durch Abstraktion und durch Aufstellen von Formeln bzw. aggregierten Ausschnittsmodellen geprägt. Die folgende Analyse geschieht dabei ‚auf Papier‘, wobei der Augenmerk auf Automatisierungs- bzw. Unterstützungsmöglichkeiten durch eine Erweiterungskomponente zu SESAM liegt. Kalibrierungsziel ist dabei Überprüfung der Spielbarkeit des Waldbrandszenarios:

- Im Waldbrandszenario ist der Ausbruch eines Feuers wie folgt definiert:
  - Zu Beginn wird die globale Temperatur auf einen Startwert z.B. 25°C gesetzt,
  - in jedem Schritt wird diese Temperatur entweder um eins erhöht oder erniedrigt,
  - falls die Temperatur über 30°C liegt, bricht mit 10% Wahrscheinlichkeit auf einem zufällig gewähltem Feld ein Feuer aus.

**Abbildung 1:** Ausschnitt aus einem Simulationslauf im Waldbrandszenario mit SESAM.



Interessant ist dabei nun die Fragestellung: Wie wahrscheinlich sind Hitzeperioden der Dauer  $t$ ? Hierfür ist es für den typischen Modellbauer (der Fachexperte) schwierig Formeln zu finden. Das Problem könnte zwar mit Randomwalks und stochastischen Prozessen gelöst werden, aber für unseren Fall genügt eine Simulation dieses Sachverhaltes.

- Ein brennendes Waldstück kann in drei verschiedenen Zuständen sein, hierbei interessiert uns hauptsächlich die durchschnittliche Dauer und die Ausbreitungswahrscheinlichkeit:
  - Schwellbrand, ca. 2 Takte: Feuer kann sich nicht ausbreiten.
  - Brand, ca. 5 Takte: Feuer kann sich in jedem Takt mit 50% Wahrscheinlichkeit ausbreiten. Ausbreiten bedeutet dabei, daß eines der drei Nachbarfelder in Windrichtung<sup>1</sup> zu brennen beginnt.
  - Lohe, ca. 3 Takte: In diesem Zustand ist ein Feuer schwer zu löschen, da es erst in einen Brand transformiert werden muß.
- Das Löschen geschieht durch Hubschrauber oder Raupen. Das Löschfahrzeug muß also zuerst zu dem Feuer fahren, um es dann löschen zu können. Unter idealisierten Bedingungen, z.B. der Wasservorrat der Fahrzeuge ist immer voll, können zwei Hubschrauber sich ein  $20 \times 30$  großes Areal aufteilen, sodaß ein Hubschrauber, falls er in der Mitte seines Areals positioniert ist, in 2-3 Takten bei einem Feuer sein kann. Da die Löschaktion einen weiteren Takt beansprucht, kann sich dieses Feuer schon zu einem Brand entwickelt haben, D.h. ein weiteres Feld kann zusätzlich brennen. Ein Hubschrauber benötigt nun zwei Takte, um zu diesem neuen Brandherd zu fliegen und zu löschen. So kann also ein neuer Brandherd, wenn er rechtzeitig erkannt wird, gelöscht werden.

<sup>1</sup> Die Windrichtung dreht sich dabei in jedem Takt um  $-1^\circ$  mit 25%,  $0^\circ$  mit 50% oder um  $1^\circ$  mit 25% Wahrscheinlichkeit.

Aus dieser Beobachtung lassen sich folgende Schlußfolgerungen ziehen. Alle Feuer können dann gelöscht werden, wenn jedes Löschfahrzeug in seinem Sektor Zeit hat, um bei Wahrnehmung eines neuen Feuers hinzufahren, es zu löschen und zum Mittelpunkt seines Areal zurückzufahren. Dies kann nur geschehen, wenn in dieser Zeitspanne kein zweites Feuer ausbricht. Die Wahrscheinlichkeit für ein Versagen eines Löschfahrzeuges kann also allgemein so beschrieben werden:

$$\begin{aligned}
 &P(\text{Innerhalb } t_f \text{ } (:= 2t_{\text{Fahren}} + t_{\text{Löschen}}) \text{ Takte nicht mehr als zwei Feuer ausbrechen)} = \\
 &P(\text{genau null Feuer in } t_f) + P(\text{genau ein Feuer in } t_f) = Bi(t_f, 0, p) + Bi(t_f, 1, p) = \\
 &\binom{t_f}{0} p^0 (1-p)^{t_f} + \binom{t_f}{1} p^1 (1-p)^{t_f-1} = (1-p)^{t_f} + p(1-p)^{t_f-1}
 \end{aligned}$$

Hierbei steht  $p$  für die Wahrscheinlichkeit eines Feuersausbruchs in seinem Areal und  $t_{\text{Fahren}}$  und  $t_{\text{Löschen}}$  für die durchschnittliche Dauer zum Hin- bzw. Wegfahren und zum Löschen und Rangieren;  $Bi$  steht dabei für die Binomialverteilung. In einem konkreten Szenario mit folgenden Werten  $t_{\text{Fahren}} \approx 3$  Takte und  $t_{\text{Löschen}} \approx 1.5$  Takte ergibt sich für  $p$ , wenn die Wahrscheinlichkeit des Ereignisses  $P$  z.B. über 90% sein soll, daß  $p \lesssim 0.016$  sein muß. D.h. die Wahrscheinlichkeit dafür, daß ein Brand pro Takt ausbrechen darf, muß unter 1.6% liegen. Dieser Wert kann nun in das Modell übernommen werden (wobei die direkte Übernahme schwierig ist, siehe den ersten Punkt), oder das Modell kann so abgeändert werden, daß innerhalb  $2 * 3 + 1.5$  Takte nur ein Feuer ausbrechen darf.

Im nächsten Abschnitt wollen wir nun klären, wie ein Programm (in unserem Fall eine Erweiterungskomponente zu SESAM) obige Analyse unterstützen kann und welche Ergebnisse wir aus diesen Betrachtungen ziehen können.

### 4.3 Ergebnisse

Eine vollautomatische Kalibrierung kommt für uns nicht in Frage, das beruht zum einen auf der fehlenden Semantik für Verhalten in SESAM und zum anderen schwerwiegenderem Teil auf der Komplexität diese Ziels. Daher werden wir – neben den vorhandenen Möglichkeiten – unterstützende Mechanismen vorschlagen, um den Modellbauer bei der Kalibrierung auf der Basis einer Makroanalyse unterstützen zu können:

- In SESAM ist es möglich, sich grafisch Parameterverläufe eines oder mehrerer Simulationsläufe anzeigen zu lassen. Eine einfache Erweiterung wäre die Möglichkeit eine vom Modellbauer definierte Kurve, die ein Makroverhalten definiert, über diese Daten zu legen, um Inkonsistenzen leicht entdecken zu können.
- Bei vielen der nächsten Punkten ist eine automatische Überprüfung des Simulationslaufes wichtig. Der Modellbauer sollte Constraints definieren können, die z.B. Relationen auf Makroebene definieren und bei deren Überprüfung Warnmeldungen während der Simulation oder in die vorhandene Logdatei geschrieben werden. In diesen Constraints kann dabei auf beliebige Parameter eines oder mehrerer Agenten oder der Umgebung zugegriffen und miteinander verglichen werden [BT94].
- Eine weiterer Mechanismus wäre eine Teilmodellsimulation, hierbei bestimmt der Modellbauer einen Teil des Gesamtmodells (z.B. nur die Feuerausbreitung ohne Einbeziehung der

Löscheinheiten) und läßt dies simulieren. Dabei kann er sich auf die dortigen Parameter beschränken, um dies zu kalibrieren. Dies ist natürlich nur in gewisser Genauigkeit möglich, da bei Mikrosimulation oft eine hohe Korrelation zwischen Parametern existiert.

- Neben dieser Teilmodellbildung kann der Modellbauer auch ein abstrakteres Modell – immer noch auf Mikroebene – bauen und die dortigen Parameter mit Parametern des ursprünglichen Modells zu verknüpfen. Z.B. kann so die Bewegung eines Löschfahrzeuges abstrakt beschrieben werden, also ohne explizite räumliche Simulation. Da das abstrakte Modell immer noch auf Mikroebene modelliert ist, können Rückschlüsse bei der Optimierung der Parameter des abstrakten Modells auf Parameter des ursprünglichen Modells gezogen werden. Dies wäre bei einem abstrakten *Makromodell* nur schwer möglich. Für diesen Punkt sind alle Unterstützungsmöglichkeiten schon in SESAM vorhanden, da ‚nur‘ ein neues Modell gebaut werden muß, die Optimierung geschieht dann wie jetzt mit Trial&Error oder mit der Erweiterung des nächsten Punktes.
- Falls genügend Rechnerkapazität vorhanden ist oder die Modelle sehr einfach sind, können mit den erwähnten Optimierungsverfahren Parameter kalibriert werden. Schwierig ist dabei allerdings die Definition der Zielfunktion (oder -relation) durch den Modellbauer, da sie auf der Mikroebene geschehen muß. Hierbei könnte eine ähnliche Vorgehensweise, wie bei den erwähnten Constraints sinnvoll sein, die statt Relation Bewertungen berechnet.
- Als letzte Erweiterung könnten noch statistische Methoden zur Verfügung gestellt werden, um Korrelationen zwischen Parameterdaten zu ermitteln. Mit diesen Informationen würde der Aufbau von Teilmodellen oder abstrakte Modellen durch den Modellbauer stark erleichtert werden. Softwaretechnisch ist es dabei am sinnvollsten das Rad nicht neu zu erfinden, sondern geeignete vorhandene Programme an SESAM zu koppeln.

## 5 Zusammenfassung und Ausblick

Die Kalibrierung von Multiagentenmodellen ist ein wichtiger Schritt bei der Modellerstellung. In heutigen Simulationswerkzeugen für Multiagentensystemen (z.B. SESAM) stehen kaum unterstützenden Mittel oder automatische Kalibrierer zur Verfügung. Anhand des Waldbrandszenarios haben wir gezeigt, wie ein Mensch mit Papier vorgehen würde, um wichtige Zusammenhänge, wie z.B. „Kann dieses Szenario überhaupt gelöst werden?“ oder konkreter „Mit welchen Parametereinstellungen ist eine Lösung möglich?“, zu ermitteln. In Kapitel 4.3 haben wir verschiedene Möglichkeiten aufgezählt, unserer Tool mit Werkzeugen zur Unterstützung des Kalibrierungsvorganges anzureichern, dabei wurden vollautomatische Methoden nicht berücksichtigt, da diese zwar ein interessantes Ziel darstellen, aber unserer Meinung nach zu komplex sind.

Nach der Erweiterung bzw. Anpassung unserer Simulationsumgebung müssen Evaluationen mit Benutzern des Waldbrandszenarios, aber insbesondere auch mit den biologischen Modellbauern unseres komplexen Bienenmodells angestrebt werden, um Aussagen über die praktische Nützlichkeit treffen zu können. Gerade für die Überprüfung bzw. den Vergleich verschiedener Modelle ist es wichtig, daß deren Parameter optimal eingestellt sind, um faire Vergleiche bzw. Überprüfungen zu ermöglichen.

## Literatur

- [BT94] BONABEAU, E. W. and G. THERAULAZ: *Why do we need artificial life?* *Artificial Life*, 1:303–325, 1994.
- [Dar94] DARLEY, VINCE: *Emergent phenomena and complexity*. In BROOKS, R. A. and P. MAES (editors): *Artificial Life IV*, pages 411–416. MIT-Press, 1994.
- [Dec96] DECKER, K.: *Distributed artificial intelligence testbeds*. In O’HARE, G.M.P. and N.R. JENNINGS (editors): *Foundations of Distributed Artificial Intelligence*, chapter 3, pages 119–138. John Wiley and Sons, 1996.
- [DKPT98] DORNHAUS, A., F. KLÜGL, F. PUPPE, and J. TAUTZ: *Task Selection in Honey Bees - Experiments Using Multi-Agent Simulation*. In WILKE, C., S. ALTMAYER, and T. MARTINETZ (editors): *Proc. of the Third German Workshop on Artificial Life, GWAL-98*, pages 171–184. Verlag Harri Deutsch AG, 1998.
- [Dor96] DORAN, J.: *Simulating societies using distributed artificial intelligence*. In TROITZSCH, K. G., U. MUELLER, G. N. GILBERT, and J. E. DORAN (editors): *Social Science Microsimulation*, chapter 17, pages 381–393. Springer, 1996.
- [DS92] DÖRNER, D. und H. SCHAUB: *Spiel und Wirklichkeit: über die Verwendung und den Nutzen computersimulierter Planspiele*. *Kölner Zeitschrift für Wirtschaft und Pädagogik*, 12:55–78, 1992.
- [Fis95] FISHWICK, P.: *Simulation Model Design and Analysis, Building Digital Worlds*. Prentice Hall, 1995.
- [FZ92] FISHWICK, P. A. and B. P. ZEIGLER: *A multimodel methodology for qualitative model engineering*. *ACM Transactions on Modeling and Computer Simulation*, 2:52–81, 1992.
- [Gol89] GOLDBERG, D.: *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, Reading, Mass., 1989.
- [Jud94] JUDSON, O.: *The rise of the individual-based model in ecology*. *Trends Ecol. Evol.*, 9(1):9–14, 1994.
- [KP98] KLÜGL, F. and F. PUPPE: *The Multi-Agent Simulation Environment SeSAM*. In BÜNING, H. KLEINE (editor): *Proc. of the Workshop Simulation and Knowledge-based Systems 1998*, 1998.
- [KPRT98] KLÜGL, F., F. PUPPE, U. RAUB, and J. TAUTZ: *Simulating multiple emergent behaviors - exemplified in an ant colony*. In ADAMI, C., R. BELEW, H. KITANO, and C. TAYLOR (editors): *Artificial Life VI*, pages 408–412. MIT Press, 1998.
- [MG98] MARCENAC, P. and S. GIROUX: *Geamas: A generic architecture for agent-oriented simulation of complex processes*. *Applied Intelligence*, 8:247–267, 1998.
- [Möh96] MÖHRING, M.: *Social science multilevel simulation with mimose*. In TROITZSCH, K. G., U. MUELLER, G. N. GILBERT, and J. E. DORAN (editors): *Social Science Microsimulation*, chapter 6, pages 123–137. Springer, 1996.
- [SZ95] SARJOUGHIAN, H. S. and B. P. ZEIGLER: *Inductive modeling: A framework marrying systems theory and non-monotonic reasoning*. In AL., P. ANTSAKLIS ET (editor): *Hybrid Systems II*, volume 999 of *Lecture Notes in Computer Science*, pages 417–435. Springer, 1995.
- [Tro96] TROITZSCH, K. G.: *Multilevel simulation*. In TROITZSCH, K. G., U. MUELLER, G. N. GILBERT, and J. E. DORAN (editors): *Social Science Microsimulation*, chapter 5, pages 107–122. Springer, 1996.