

Evaluation von Entwicklungsumgebungen für Multi-Agenten-Simulationen

Marco Kerwitz¹, Michael Clasen¹

Abstract: Multi-agent-simulations are gaining more and more importance because of the growing desire to do forecasts and analysis before conducting projects and experiments. Resulting from the rising popularity of this topic, corresponding development frameworks and tools have been build and made available to the general public. Due to the wide-spread deployment of these tools it is generally not easy for the enduser to gather enough information to compare the available MAS-programs. In this paper we will give an overview of available tools and make a suggestion.

Keywords: Multi-Agenten-Simulation, Industrie 4.0

1 Bedeutung von Multi-Agenten-Simulationen

Multi-Agenten-Simulationen (MAS) werden seit einigen Jahren erfolgreich eingesetzt, um Problemstellungen der unterschiedlichsten Forschungsbereiche zu simulieren [Ra02]. Zu den bekanntesten Arbeiten zählen sicherlich die Sugarscape-Modelle von Epstein und Axtell [EA96], mit denen eine Vielzahl sozialer Phänomene nachgebildet und z.T. erklärt werden konnten. Aber auch im Bereich der Agrarwissenschaften finden sich einige wenige Arbeiten, in denen Multi-Agenten-Simulationen zum Einsatz kamen [z.B. Be04, Mü08]. Vor dem Advent der Industrie 4.0- und Farming 4.0-Ansätze ist jedoch mit einer Renaissance der MAS zu rechnen. Zentrales Element dieser Ansätze ist die autonome Steuerung von Produkten oder Maschinen, die selbsttätig nach Entscheidungsregeln vorgegebene Ziele verfolgen [Cl16]. Der Erfolg hängt dabei maßgeblich von der Güte der Entscheidungsregeln ab. Um ein System von Beginn an mit guten Entscheidungsregeln starten zu lassen, müssen diese Regeln zuvor in Simulationen getestet werden. Auch können MAS als Basis für die Steuerung der realen Systeme dienen. Gründe genug, sich einen aktuellen Überblick [Cl02] über die derzeit verfügbaren Werkzeuge zur MAS zu verschaffen.

2 Auswahl, Bewertung und Klassifikation der Tools

Zum Aufspüren möglichst vieler MAS-Tools wurde zunächst eine Liste mit 34 Suchbegriffen, wie beispielsweise „MAS-Tool“ oder „Cellular Automata“, zusammengetragen. Nach diesen Begriffen wurde per Google Web Search und Google

¹ Hochschule Hannover, Fakultät IV – Wirtschaft und Informatik, Ricklinger Stadtweg 120, 30459 Hannover, marco@kerwitz.com, michael.clasen@hs-hannover.de

Scholar, sowie in Google Books, Apple iBooks und den Katalogen der Hochschulbibliothek der Hochschule Hannover gesucht. Das Ergebnis war eine Liste mit 102 Werkzeugen, die Simulationsaufgaben unterstützen.

Alle 102 Werkzeuge wurden dann anhand der Kriterien „Tauglichkeit für ökonomische Fragestellungen“, „Verfügbarkeit (grundsätzliche Bezugsmöglichkeit, Preis)“, „Dokumentation“, „Einarbeitungsaufwand“, „Aktualität (Datum des letzten Updates)“ und „Popularität (Anzahl der Projekte, Größe der Community)“ mit einem Wert von 0-2 bewertet. Eine 0 steht dabei für „nicht geeignet“ (bzw. „nicht verfügbar“, „nicht aktuell“ etc.), eine 1 für grundsätzlich geeignet und eine 2 für gut geeignet. Tauglichkeit und Verfügbarkeit waren Totschlagargumente, so dass Tools, die hier mit 0 bewertet wurden, nicht weiter betrachtet worden sind. Die 7 Tools mit der höchsten aufsummierten Bewertung wurden dann in einem zweiten Schritt näher untersucht. Hierzu wurden drei Simulationsaufgaben mit unterschiedlichem Schwierigkeitsgrad definiert, die mit den 7 Werkzeugen gelöst werden sollten. Diese waren ein einfaches Predator & Prey-Modell, bei dem die Populationsentwicklung von Räuber und Beute analysiert wird, ein Modell das parkplatzsuchende PKW simuliert und ein Marktmodell, indem der Wohnungsmarkt einer Region analysiert werden sollte.

3 Vorstellung empfehlenswerter Tools

Die Tabelle gibt einen Überblick über die empfohlenen Tools. Im Folgenden werden diese vorgestellt und auf ihre Vor- und Nachteile eingegangen.

Name	Entwickler	Lizenz	Bezugsquelle
Anylogic	The Anylogic Company	Kostenpflichtig, kostenlose Testversion	www.anylogic.de
SeSAM	Universität Würzburg	LGPL, Open Source	130.243.124.21/sesam
MadKit	Univsit� Montpellier	kostenlos, Open Source	www.madkit.net
NetLogo	CCL	kostenlos, GNU	ccl.northwestern.edu/netlogo
FlameGPU	University of Sheffield	kostenlos	www.flamegpu.com

Tab. 1: Übersicht über die analysierten Tools

Mit **AnyLogic** lassen sich sehr komplexe Simulationen programmieren und zwei- oder dreidimensional darstellen. Agenten, Objekte und die grafische Darstellungsfläche lassen sich ohne Programmierung einfach per Mausklick erstellen, konfigurieren und steuern. Die Aktionen der Agenten können per Flowchart mithilfe verketteter Regeln angelegt werden, so dass für einfache Simulationen keine Java-Kenntnisse nötig sind. Auch Diagramme, die zur Laufzeit Informationen über die Simulation liefern, sind mit wenigen Klicks erzeugt. Positiv ist weiterhin, dass AnyLogic eine große Anzahl an Bausteinen bereitstellt, die per Drag & Drop in das Modell integriert werden können. Für die Einarbeitung in AnyLogic stehen diverse Beispielmuster, einige Tutorials und ein

umfangreiches Handbuch im Internet frei zur Verfügung. Die Beispielsimulationen verringern zwar die ersten Hürden, doch es braucht Zeit, sich in die vielen Features einzuarbeiten. AnyLogic ist ein sehr flexibles MAS-Programm, mit dem sich auch komplizierte Simulationen realisieren lassen. Aufgrund seiner hohen Flexibilität ist AnyLogic für nahezu jedes Simulationsproblem geeignet. Lediglich bei sehr einfachen Simulationen gilt es abzuwägen, ob die hiermit verbundene Komplexität den Einarbeitungsaufwand rechtfertigt.

SeSAM könnte als die OpenSource Version von AnyLogic bezeichnet werden. Bedienung und Funktionsumfang von SeSAM (Shell for Simulated Agent Systems) ist somit grundsätzlich mit dem von Anylogic vergleichbar, jedoch macht SeSAM einen rustikalen und etwas umständlicheren Eindruck. Die Programmierung von Simulationsmodellen kann auch bei SeSAM per Mausklick erfolgen. Regelketten, die das Handeln eines Agenten bestimmen, werden direkt in der sogenannten „Reasoning Engine“ in einem Flowchart erzeugt, was das Verständnis der Vorgänge stark vereinfacht. Die Regeln bestehen zwar aus verketteten Java-Methoden, aber auch diese kann der Nutzer mit einem Assistenten zusammensetzen und konfigurieren. Größere MAS, die mehrere Durchgänge mit verschiedenen Startparametern umfassen, unterstützt das Tool mit einem Assistenten und der Möglichkeit, Durchgänge im lokalen Netzwerk zu verteilen. Nachteilig fielen lediglich die fehlende Möglichkeit für dreidimensionale Simulationen sowie die gegenüber AnyLogic deutlich kleinere Bibliothek an vorgefertigten Programmbausteinen auf. Um die Einarbeitungszeit zu erleichtern, werden über ein Community Wiki neben einer Dokumentation der im Programm enthaltenen Features auch diverse Anleitungen, Beispielmodelle und nützliche Tutorials bereitgestellt.

MaDKit ist eine Open-Source Plattform für die Entwicklung von Java basierten MAS-Anwendungen und Simulationen, die sich, anders als üblich, nicht auf einzelne Agenten, sondern auf Organisationen konzentrieren. Der Ansatz, den MaDKit dabei verfolgt heißt AGR (Agent / Gruppe / Rolle) und bedeutet, dass einzelne Agenten bestimmte Rollen in einer Gruppe übernehmen und dadurch künstliche Populationen bilden. Die Macher verzichten ganz bewusst auf eine standardisierte Basisklasse von Agenten, um für eine ungleichmäßige Masse zu sorgen. Gleichwohl stellen sie dem Nutzer grundlegende Funktionalitäten für die Kommunikation unter den Agenten zur Verfügung. Für die Entwicklung mit MaDKit empfiehlt sich die Eclipse IDE, die auch in den Anleitungen (<http://www.madkit.org/madkit/tutorials>) immer wieder Erwähnung findet. Durch seine Fokussierung auf die Simulation von Populationen und Gesellschaften eignet sich die Plattform eher für größere Mengen von Agenten, beispielsweise bei der Ausbreitung eines Krankheitserregers, als für einfache Simulationen.

NetLogo ist eine Programmiersprache mit integrierter Programmierumgebung für Multi-Agenten-Simulationen. Sie ist die Weiterentwicklung der Programmiersprachen Logo und StarLogo. Hauptziel dieser Sprachen ist eine leichte Erlernbarkeit, so dass sich diese besonders gut für Programmieranfänger und für Schulungszwecke eignen. Hierunter leidet aber die Leistungsfähigkeit. Zwar wird das Simulationsgeschehen auch hier zur

Laufzeit graphisch aufbereitet, bei komplexen Problemstellungen stößt NetLogo aber schnell an seine Grenzen. Die Einarbeitung in NetLogo wird durch die zahlreiche Beispiele und ein Dictionary auf der NetLogo-Website, sowie durch diverse Video-Tutorials auf externen Webseiten leicht gemacht.

FlameGPU wurde entwickelt, um eine sehr große Anzahl von Agenten sowie deren Verhalten und Eigenschaften zu berechnen und grafisch aufwändig darstellen zu können. Das Tool ist eine Erweiterung des CPU-basierten FLAME Frameworks und stellt eine einfache Möglichkeit dar, Agenten konventionell in der Programmiersprache C zu programmieren und trotzdem die sehr performante CUDA-Technik zu verwenden. CUDA-Code wird statt auf der CPU auf der GPU, also dem Grafikchip, ausgeführt, was erhebliche Verbesserungen bei der Berechnung von komplexen Modellen durch Parallelisierung bietet. Der Vorteil für den Anwender liegt dabei in der Abstraktion: er kann sich auf die Beschreibung der Agenten konzentrieren und die deutlich bessere Performance der GPU nutzen, ohne Experte für CUDA-Programmierung sein zu müssen. Ein weiterer Vorteil dieser Herangehensweise ist die Nähe der berechneten Daten zum Grafikchip: eine Echtzeitbetrachtung von riesigen Populationen ist möglich, weil sich die Daten bereits auf der darstellenden Hardware befinden. Für die Benutzung setzt FlameGPU das Visual Studio von Microsoft voraus und die Definition von Agenten erfolgt im XML-Schema. Die Anbieter von FlameGPU bieten auf der Homepage des Projekts eine umfangreiche Dokumentation und ein Verzeichnis von Veröffentlichungen über das Tool an, was den Einstieg und die Arbeit einfacher gestaltet.

Literaturverzeichnis

- [Be04] Berger, T. (2004), Agentenbasierte Modellierung von Landnutzungsdynamiken und Politikoptionen, in: *Agrarwirtschaft* 53 (2), 77-87.
- [Cl02] Clasen, M. (2002): Entwicklungsumgebungen für Multi-Agenten-Systeme zur Simulation von digitalen Marktplätzen. in: Wild, K., Müller, R. A. E., Birkner, U. (Hrsg): *Referate der 23. GIL-Jahrestagung in Dresden 2002*, Dresden, S. 34-37.
- [Cl16] Clasen, M. (2016): Farming 4.0 und andere Anwendungen des Internet der Dinge. in: *dieser Tagungsband*.
- [EA96] Epstein, J. M., Axtell, R. (1996): *Growing Artificial Societies. Social Science from the Bottom up*, Washington.
- [Mü08] Müller-Scheeßel, J. (2008): *Ökonomische Analysen zur Anbaubereitschaft und zur rechtlichen Ausgestaltung des Anbaus gentechnisch veränderten Rapses*. Wissenschaftsverlag Vauk, Kiel.
- [Ra02] Rauch, J. (2002): *Seeing Around Corners*, The Atlantic Online, <http://www.theatlantic.com/issues/2002/04/rauch.htm>, Stand: 13.06.02