Cynthia Nikolai and Gregory Madey (2009)

Tools of the Trade: A Survey of Various Agent Based Modeling Platforms

Journal of Artificial Societies and Social Simulation vol. 12, no. 2 2
<http://jasss.soc.surrey.ac.uk/12/2/2.html>

For information about citing this article, click here

Received: 29-Jul-2008 Accepted: 05-Mar-2009 Published: 31-Mar-2009

Abstract

Agent Based Modeling (ABM) toolkits are as diverse as the community of people who use them. With so many toolkits available, the choice of which one is best suited for a project is left to word of mouth, past experiences in using particular toolkits and toolkit publicity. This is especially troublesome for projects that require specialization. Rather than using toolkits that are the most publicized but are designed for general projects, using this paper, one will be able to choose a toolkit that already exists and that may be built especially for one's particular domain and specialized needs. In this paper, we examine the entire continuum of agent based toolkits. We characterize each based on 5 important characteristics users consider when choosing a toolkit, and then we categorize the characteristics into user-friendly taxonomies that aid in rapid indexing and easy reference.

Keywords:
Agent Based Modeling, Individual Based Model, Multi Agent Systems

Introduction

1.1

In the past few years, several seminal ABM surveys have emerged. They are a giant stride in the right direction, but current surveys generally are limited to four or five mainstay and characteristically or historically similar toolkits (Castle 2006; Railsback 2006; Tobias 2004). Moreover, these surveys are presented from the point of view and for the intended audience of one or two communities of interest (Castle 2006; Railsback 2006; Serenko 2002; Tobias 2004). However, different groups of users prefer different and sometimes conflicting aspects of a toolkit. For example, social scientists, who may have little or no programming experience, are concerned more with ease of use, the degree of programming skills required, and the inclusion of intuitive interfaces to manage simulations. Many, in general, are not concerned about whether the software is open source or restricted open source. To computer scientists, however, the type of license that governs the toolkit is a big consideration; they want the ability to "get behind the scenes" of a toolkit and to have the programming flexibility to modify or extend the software with third party applications if necessary. They also generally prefer saving execution time by programming simulations themselves rather than using built-in interfaces, which usually are less computationally efficient. Teachers of ABM, on the other hand, want packages that are easy to learn, that offer pedagogical insights, and that provide the student with the ability to transition to more difficult and comprehensive toolkits in the future.

1.2

In this paper we address the issues of the broader ABM community. This paper is a survey of the toolkits that are available today and how they compare to each other from an interdisciplinary and multi-stakeholder perspective. Our goal is to provide users with the ability to better choose a suitable toolkit based on the features abstracted from various documentation and compiled into an easy to use compendium. In addition, we expand the ABM body of knowledge to include information about a breadth of characteristically and historically diverse platforms.

1.3

This work is the result of ongoing research into various characteristics of ABM toolkits. In this paper, we examine 5 characteristics across the spectrum of toolkits: programming language required to create a model or simulation, operating system required to run the toolkit, type of license governing the platform, primary domain for which the toolkit is intended, and degree of support available to the user of the toolkit.

1.4

This paper is structured as follows. First, we begin with some limitations of current ABM surveys. This is followed by a comparison of the characteristics of various toolkits in the form of tabular taxonomies followed by a text explanation. Finally, we conclude the paper with a full representation of features for each toolkit in a quick and easy to use matrix format.

Related Work

2.1

In the last few years, the ABM community has made giant strides in developing practical agent based modeling toolkits that enable individuals to develop significantly sized applications. More and more such toolkits are coming into existence, and each toolkit has a variety of characteristics. Some toolkits are built for general purpose modeling and some are built for a particular domain. Some are open source, some are closed source, and others are proprietary. Some toolkits have a simple user interface, and others require complex programming techniques. Several individuals have made attempts to compare toolkits to each other. One of the seminal papers has been the investigation by Railsback, Lytinen, and Jackson (2006). In this paper, the authors examine four main platforms: NetLogo, Mason, Repast, and Swarm. They create a template, called a "StupidModel," for various levels from which to evaluate and compare toolkits to each other. For example, for level 1, they examine the underlying environment and how various toolkits display agents in their environment. With each new level, they add more capabilities and examine how each toolkit compares to the others. For level 2, they add more agent actions and examine how different platforms implement scheduling for these actions. They continue adding more capabilities through 15 different levels, through which they examine characteristics such as environmental issues, model structure, agent scheduling, file input and output, random number generation, and statistical capabilities.

2.2

This survey is a great step in the right direction, but the main limitation is that it only examines 4 platforms. In addition, most of the toolkits are historically similar in nature. Even to the extent that Swarm and Mason were designed as general purpose toolkits, Repast was designed for social scientific use, and NetLogo was intended as an educational tool, three of the toolkits are descendants of Swarm, while one is descended from an educational lineage. Our work differs from Railsback et al, in three main respects. First, we expand the ABM body. We consider not only general or mainstay toolkits from the same lineage, but we also consider less well known and diverse specialized platforms as well. A second difference is that in this paper, we do not evaluate toolkits as better or worse than others. Our goal is to present the facts and to let the reader choose which toolkit is the most suitable match for his/her project. In future continuation of this work, we hope to "get under the hood" and do a more comparative study of the toolkits. Finally, whereas Railsback et al evaluate four toolkits in depth, this work only scratches the surface of the toolkits. For now, we only examine 5 characteristics that individuals examine when attempting to choose a toolkit for their project.

2.3

The second main ABM survey is by Castle and Crooks (2006). In this paper, Castle and Crooks examine 8 simulation platforms: Swarm, Mason, Repast, StarLogo, NetLogo, Obeus, AgentSheets, and AnyLogic. They have a particular focus on evaluating geospatial capabilities. They also address several additional characteristics including date of inception, implementation language, required programming experience, and availability of demonstration models and tutorials.

2.4

Again the main limitation is that the study only examines a handful of the ABM toolkits that are available. The main audience is the domain of geospatial modeling, and again, it compares only general purpose, characteristically similar toolkits and toolkits specialized for the social sciences. In our work in this paper, we expand the ABM knowledge base to incorporate a more diverse and expansive continuum of toolkits. We also examine several characteristics in more detail. In addition, we facilitate toolkit selection by including not only comparisons of characteristics across toolkits, but we also include matrices comparing toolkits across characteristics.

2.5

The third survey is by Tobias and Hofmann (2004). In this survey, the authors examine 4 main open source toolkits: Repast, Swarm, Quicksilver, and VSEit, and they evaluate them based on various types of criteria, to include general criteria, modeling and experimentation, support for modeling, and modeling options. Altogether, they examine 19 different characteristics across these 4 platforms. Next they rank the platforms by assigning scores to represent the quality of the criteria of interest. The paper examines a broad range of characteristics, and this is what we hope to model in our future toolkit research. The main limitation of this survey is that it is from the point of view of social scientists, and it only examines "free" libraries in use by the social scientific community that use Java as the main programming language. With our work, on the other hand, we hope to appeal to the broader ABM community. We also bring to the fore additional toolkits that are geared toward the social sciences, both in general and in particular specializations.

2.6

A fourth survey paper on agent based toolkits is by Serenko et al (2002). In this work, the authors investigate 20 toolkits from an educational perspective based on their use as pedagogical tools in post-secondary courses. They classify toolkits based on 4 characteristics, namely, ability to create mobile agents, ability to develop a multi-agent system, ability to create different kinds of agents for different purposes (effectively, agent based toolkits), and ability to retrieve information. They also examine the underlying language required for programming a model or simulation. Next they interview 87 instructors who are using these toolkits and who evaluate the toolkits based on user satisfaction with platform functionality, performance, and user interaction. This is a good attempt to compare a breadth of the agent toolkits across multiple characteristics. Our work in this paper is similar in that we attempt to survey the breadth of available toolkits. However, we examine toolkits from a multi-stakeholder perspective. We also investigate more objective characteristics. In the future, we hope to implement a questionnaire to survey similar advanced characteristics.

Limitations

3.1

Before we delve too far into this paper, we would like to underscore a few limitations. The major limitation of this survey is its scope. We chose to examine a large breadth of platforms across a small range of characteristics. This has two important implications. First, we are not able to evaluate the depth of the platforms in terms of all of their characteristics. Second, we are not able to examine a wide berth of characteristics. While this is a good base of potential characteristics of interest, certainly, there are additional characteristics that are important factors in one's decision to choose one platform over another. In our ongoing research into ABM toolkits, we will examine more in-depth and complex characteristics.

3.2

Another limitation of this survey is the disparity in degree of documentation for various toolkits. Some platforms are widely in use and have ample documentation, and other platforms have barely any documentation. Even so, we tried to look at each platform in an equal manner. However, we were limited to what we were able to find on the internet and in journals. In addition, we tried to examine each characteristic as completely and comprehensively as possible. However, our study is not complete. In places where it is not complete, it is because the developers have not specified the complete granularity of the platforms with respect to the characteristics evaluated.

3.3

Third, there is a disparity in the quality of documentation that is included with each platform. Documentation ranges from very detailed to hardly any details at all. In this survey, we do not attempt to evaluate the quality of the documentation. Rather, we try to classify the toolkits based on the types of documentation that are available to support the user.

3.4

Finally, another challenge to this study is the conflicting use of terms in different domains. Since the agent based modeling field has developed from multiple disciplines (e.g. social science, artificial intelligence, and computer science), many of the terms are not used consistently across various domains . For example, the three most inconsistent terms are "agent," "agent-based," and "multi-agent." When toolkits from different domains use the term multi-agent system, it is unclear if they mean a system capable of modeling a large number of fairly homogenous agents (agent based system) or a smaller system of heterogeneous agents equipped with artificial intelligence (true multi-agent system). We do not attempt to disambiguate the terms for each of the fields in this paper. Rather, we attempt to examine the overall domain from an agent based perspective (as opposed to a multi-agent system perspective). Because of the inconsistent use of terms, it was difficult to classify the toolkits into precise taxonomies.

Methodology

4.1

We began this survey by gathering a comprehensive list of agent based toolkits available and that are being used in some fashion for ABM purposes. These include any platforms that are available in the public domain, including open source and closed source, general purpose and specialized, as well as free and proprietary toolkits. We tried to make this as comprehensive as possible. Next, we gathered as much information as we could from open sources. We scoured white papers, technical papers, journals, and various websites to gather as much information as possible. Where there was third hand information, we confirmed it by going directly to the source. Next, we sorted through all the information and created various taxonomies based on major classifications. Based on the taxonomies, we created corresponding tables that allow individuals to quickly compare various toolkits based on particular characteristics of interest. The following toolkits were considered:

- Agent Building and Learning Environment (ABLE)
- AgentBuilder Lite/Pro
- Tryllian Agent Development Kit (ADK)
- AgentSheets
- AnyLogic
- Ascape
- Brahms
- Breve
- Common-pool Resources and Multi-Agent Systems (Cormas)
- Cougaar
- DeX
- Distributed operator model architecture (DOMAR)
- ECHO
- jEcho
- ECJ
- iGen
- JADE

- JAS
- Java Auction Simulator API (JASA)
- JCA-Sim
- Java Enterprise Simulator (jES)
- JESS
- Laboratory for Simulation Development (LSD)
- Multi Agent Development Kit (Madkit)
- Rules Based Multi-Agent System (MAGSY)
- Multi-agent modeling language (MAML)
- Mason
- Multi-Agent Simulations for the SOCial Sciences (MAS-SOC)
- Matrix Laboratory (Matlab)
- Micro-und Multilevel Modelling Software (MIMOSE)
- Moduleco
- NetLogo
- Object Based Environment for Urban Simulation (OBEUS)
- oRIS
- Political Science- Identity (PS-I)
- Framework for Agent-based Modelling with Java (FAMOJA)
- Quicksilver (now called omonia)
- REcursive Porous Agent Simulation Toolkit (Repast) and family (e.g. RepastS, RepastPy, RepastJ, Repast.net)
- Strictly Declarative Modeling Language (SDML)
- Jade's sim++
- SimPlusPlus
- SimAgent (aka sim_agent)
- SimBioSys
- Multimodeling Object-Oriented Simulation Environment (Moose)
- SimPack
- Spatial Modeling Environment (SME)
- Shell for Simulated Agent Systems (SeSAm)
- SOAR
- StarLogo and family (e.g. StarLogo T, StarLogo TNG, OpenStarLogo, MacStarLogo)
- Sugarscape
- Swarm
- Versatile Simulation Environment for the Internet (VSEit)
- Zeus

Results

5.1 The five characteristics we examine in this paper are: language required to program a model and to run a simulation, operating system required to run the toolkit, type of license that governs the toolkit, primary domain for which the toolkit is intended, and types of support available to the user. We chose these characteristics because they are usually the first features that one looks at when choosing a toolkit for a project (Castle 2006; Leszczyna 2004; Tobias 2004). We have two main results. First, we define taxonomies that allow for easy comparison of one characteristic across all of the platforms. This enables one to select candidate toolkits from across the ABM spectrum based on one or two characteristics of interest. Second, we define a matrix that shows in a condensed form all of the characteristics of interest across one platform. This helps one to see how a particular platform measures up as a whole for each of the characteristics across one's needs.

Programming Language

5.2 There are various programming languages that may be used to program an agent based model and to run a simulation. Programming languages are important because different languages have different implications in terms of ease of programming, portability, and compatibility. The main programming languages used across the ABM spectrum are summarized below. Note that these are the languages that are used to *program a model* using the toolkit rather than the underlying languages that are used to create the toolkit.

- Able Rule Language (ARL)
- Any language that supports activeX components (e.g. C, C++, VB, VBA, Java)
- All languages that are compiled into Java or scripting languages which are executed in the Java Virtual Machine
- BeanShell (Java interpreted)
- Brahms language (an agent oriented language)
- C
- C++
- Cellular Description Language (CDL) (for input to simulation)
- COGNET Execution Language (CEL)
- dML (deX Modeling Language): a domain-specific language based on C++
- AgentSpeak(XL), an extension of Agentspeak(L) and (Environment Description Language for Multi-Agent Simulation) ELMS, a language for modelling environments where cognitive agents are situated
- Java
- Jess (rule based language)
- UML-RT (UML for real time)
- Knowledge Query and Manipulation Language (KQML)

- Lisp
- Logo
- NetLogo
- MacStarLogo
- StarLogo (an extension of Logo)
- StarLogoT
- StarLogo TNG
- LSD (functional language derivative)
- Magsy (production language)
- Multi-agent Modeling Language (MAML)
- MATLAB®
- Microsoft.net .NET languages (C#, C++, Visual Basic, .Net, J#)
- Model Description Language (derived from functional language paradigms)
- Objective C
- Oris (dynamic and interpreted multi-agent language very close to C++ and Java)
- Pop-11 (similar to common lisp)
- Prolog
- Python
- Scheme (Kawa)
- SeSAm-Impl and SeSAm-UML
- Smalltalk
- Standard ML
- StarLogo TNG visual programming language
- Steve (a simple interpreted object oriented language)
- Tcl/tk scripting
- Visual programming
- Visual AgentTalk (VAT)

5.3

Agent models can be programmed in virtually all of the main programming languages, including C, C++, and Java. These are mainly used for the toolkits that are designed for general purposes. The rest of the languages are languages that stem from a need for specialization. Most languages in specialized toolkits are created and used specifically for that toolkit. We can also see in this table a little of the direction/roots of the languages. For example, we see a small lineage forming around the Logo language. That is, NetLogo, MacStarLogo, StarLogo, StarLogoT, and StarLogo TNG are derivatives of Logo.

Platforms Per Subcategory

5.4

By far, the main programming language most models have adopted is Java. About 42% of the platforms employ Java as their primary programming language. Toolkits that support Java programming are listed below.

- ADK
- AgentBuilder
- AnyLogic
- Ascape
- Cougaar
- DOMAR-J
- jEcho
- ECJ
- FAMOJA
- iGen
- Jade
- JAS
- JASA
- JCA-Sim
- jES
- JESS
- Madkit
- Mason
- Moduleco
- Omonia
- RepastS
- RepastJ
- SimPack
- SOAR6
- Sugarscape
- Swarm
- VSEit

5.5

The next three largest contingents are C, C++, and the Logo dialects. About 17% of the platforms use C++ to program models, about 11% use C, and about 8% use a variant of Logo. Approximately 28% of the toolkits use a platform specific language which the toolkit authors designed to facilitate programming models and simulations in that domain. Note that the sum of this collective is above 100%. This is because several platforms support multiple languages. We will begin with toolkits that support C++ programming. These are:

- AgentBuilder
- DeX

- iGen
- LSD
- Madkit
- MOOSE
- Jade's Sim++
- SimBioSys
- SimPack
- SOAR6

5.6

Next, we have toolkits that support C programming.

- AgentBuilder
- Echo
- iGen
- Madkit
- MAML
- SimPack (no longer maintained)
- SOAR6

5.7

Third, we have toolkits that support programming in Logo Dialects.

- NetLogo
- MacStarLogo
- StarLogo
- OpenStarLogo
- StarLogoT
- StarLogo TNG

5.8

Finally, we have toolkits that support visual programming (table 1). That is, these platforms have graphical-based programming capabilities that generally are much more simple to learn and use than traditional programming languages. In the future work, we would like to examine further the extent to which toolkits have visual programming capability in addition to programming language capability.

Table 1: Toolkits That Support Visual Programming

Visual Programming Language	**Toolkit**
General visual programming	MAML PS-I RepastS SeSAm SimPlusPlus SME Zeus
StarLogo TNG visual programming language	StarLogo TNG
Visual AgentTalk (VAT)	AgentSheets

5.9

Table 2 depicts the remaining domain of toolkits per programming language.

Table 2: Remaining Toolkits per Programming Language

Programming Language	**Toolkit**
Able Rule Language (ARL)	ABLE
Any language that supports activeX components (e.g. C, C++, VB, VBA, Java)	SimPlusPlus
All languages that are compiled into Java or scripting languages which are executed in the Java Virtual Machine	Madkit
BeanShell (Java interpreted)	Madkit
Brahms language (an agent oriented language)	Brahms
Cellular Description Language (CDL) (for input to simulation)	JCA-Sim
COGNET Execution Language (CEL)	iGen
dML (deX Modeling Language): a domain-specific language based on C++	DeX
AgentSpeak(XL), an extension of Agentspeak(L) and (Environment Description Language for Multi-Agent Simulation) ELMS, a language for modelling environments where cognitive agents are situated	MAS-SOC
Jess (rule based language)	JESS Madkit
UML-RT (UML for real time)	AnyLogic
Knowledge Query and Manipulation Language (KQML)	AgentBuilder
Lisp	OMAR-L

	SimAgent SOAR1-5
LSD (functional language derivative)	LSD
Magsy (production language)	MAGSY
Multi-agent Modeling Language (MAML)	MAML
MATLAB®	Matlab
Microsoft.net .NET languages (C#, C++, Visual Basic, .Net, J#)	OBEUS Repast.net
Model Description Language (derived from functional language paradigms)	MIMOSE
Objective C	Swarm
Oris (dynamic and interpreted multi-agent language very close to C++ and Java)	oRIS
Pop-11 (similar to common lisp)	SimAgent
Prolog	SimAgent
Python	Breve DeX Madkit RepastPy
Scheme (Kawa)	Madkit
SeSAm-Impl and SeSAm-UML	SeSAm
Smalltalk	Cormas SDML
Standard ML	SimAgent
Steve (a simple interpreted object oriented language)	Breve
Tcl/tk scripting	PS-I (only to apply affects) SOAR

Type of License

5.10
The main domain of licenses that governs various toolkits is depicted below.

- Associated third party licenses (usually non-proprietary)
- Contact authors for availability
- Contact Tryllian to acquire a closed source license
- Free
    - Open Source
        - Academic Free License
        - Artistic License Agreement
        - BSD
        - Cougaar Open Source License (COSL)
        - GPL
        - LGPL
    - Closed Source
    - Conditionally Free
    - Academic Purposes
    - To modify but not to distribute the modified version
    - Use and distribution for non-commercial purposes
    - See license for details
- Proprietary
- Discounted
    - Academic License

5.11
The type of license is important because it has implications for releasing the source code under commercial distribution. For example, for platforms licensed under the GNU Lesser General Public License (LGPL), if one wants to release a modified version of the toolkit for commercial purposes, one also has to release the source code of the modified platform (GNU Website). Toolkits licensed under the Berkley Software Distribution (BSD) license, on the other hand, do not require one to release the source code of commercial extensions to the platform (freebsd.org website).

5.12
We have organized the licenses into four main branches. We can see that the majority of the toolkits are free (about 76%). These are broken down further into open source (about 53%), closed source (about 9%), and free with restrictions (14%). Of the remaining toolkits, about 17% are proprietary. The last 5% are available under contract through case by case arrangements with the authors. Finally, in addition to regular licenses, some of the toolkits come with associated third party licenses for software that is already incorporated into the toolkit or for additional features that may be incorporated into the toolkit by the user.

Platforms Per Subcategory

5.13

We begin with free toolkits. As depicted in list 6 above, we have free open source, free closed source, and conditionally free toolkits. Free open source toolkits release the source code with their toolkit and allow modifications in accordance with their governing license. Common open source licenses include Berkley Software Distribution (BSD), GNU General Public License (GPL), GNU Lesser General Public License (LGPL), and the Cougaar Open Source License. Free closed source toolkits, on the other hand, do not release the source code to the public. Finally, we have conditionally free licenses. These toolkits are free, but they have conditions on how they are used. For example, some licenses are free if they are used only for academic purposes. Others are free as long as they are used for non-commercial purposes. These licenses are mostly closed source. Proprietary toolkits, on the other hand, require the user to pay the toolkit authors for a license. Finally, some toolkit authors will negotiate licenses with the users according to the circumstances or intended purposes of the user. These fall under "contact authors for availability." The toolkits classified as free under open source are shown in table 3 below.

Table 3: List of Toolkits Classified as Free Under Open Source Licenses

Type of License	**Toolkit**
Open Source (uncategorized[1])	ABLE (for academic and non-commercial use) DeX DOMAR ECHO jECHO MAML (for evaluation purposes) SimAgent Zeus (read license)
Academic Free License	ECJ jES Mason
Artistic License Agreement	SimBioSys
BSD	Ascape Repast (RepastJ, RepastPy, RepastS, Repast.net) SOAR
Cougaar Open Source License (COSL)	Cougaar
GPL	Breve JASA LSD Madkit (for development and non-commercial applications) Moduleco PS-I Jade's sim++ SDML SimPlusPlus SimPack SME Sugarscape Swarm
LGPL	ADK iGen Jade JAS Madkit (for basic libraries) FAMOJA Omonia SeSAm

[1] These toolkits do not fall under standard licensing agreements such as BSD, GPL, and LGPL. The licensing generally is defined by the authors/developers of the toolkits

5.14

The following toolkits do not release the source code.

- Brahms
- JCA-Sim
- MAGSY
- MacStarLogo
- MIMOSE
- NetLogo
- OBEUS
- StarLogo
- StartLogo T
- StarLogo TNG
- VSEit

5.15

The list below shows the toolkits that have proprietary licenses. Note that some of these toolkits are free or discounted if they are used solely for academic purposes.

- AgentBuilder (discounted academic licenses)
- AgentSheets
- AnyLogic
- iGen (discounted academic licenses)
- JESS (free for academic purposes)
- Matlab
- oRIS (free for academic purposes)

5.16

Finally, in table 4, we see the spectrum of toolkits that are free under certain restrictions. For example, some platforms are free to use and distribute as long as they are used for solely non-commercial purposes. Others are free as long as they are used for academic purposes. Another category of toolkits is governed by licenses that restrict individuals from distributing modified versions of the source code. Finally, some platforms have their own unique/hybrid licenses that are best suited for user to view for himself/herself.

Table 4: Toolkits That Are Free Under Certain Restrictions

Type of License	Toolkit
Academic Purposes	ABLE Brahms (closed source) JESS oRIS SDML
To modify but not to distribute the modified version	Cormas
Use and distribution for non-commercial purposes	ABLE OpenStarLogo
See license for details	SimAgent

5.17

The majority of the remaining toolkits use their own special purpose licenses. These toolkits are depicted in table 5 below.

Table 5: Licenses Employed by Various ABM Toolkits

Type of License	Toolkit
Associated third party licenses (usually non-proprietary)	MAS-SOC SDML SimAgent
Contact authors for availability	MAS-SOC
Contact Tryllian to acquire a closed source license	ADK

Operating System

5.18

The third category we examine is the operating system on which the toolkits run. The operating system domain is depicted as follows. Note the variety of operating system specifications defined in the literature.

- ADUX
- AIX
- Any platform that supports C++/any C++ compiler
- Any platform with a Java Virtual Machine (JVM)
  - Java 2 JVM
  - SDK version 1.4.1 or later
  - Java 2 SDK
  - Java Runtime Environment (JRE) 1.5.0 or later
  - Java SDK 5.0 or better
  - JDK 1.1
  - Any Java Development Kit (JDK) installation
  - Java 2 Runtime Environment(JRE) and Internet Explorer 5.x or greater
  - JRE Java version 1.4
  - JRE version 1.3.1
  - Bea's JRockit JVM
  - IBM's JVM
- BSD
- DOS
- Emulation of Windows NT or Linux
- HPUX
- IA32 Linux; PPC Linux

- IBM mainframes
- Java-1.4-capable PDAs
- Linux
    - x86 or x86_64 linux
    - SuSE Open Linux 10.2 or later, x86-32
    - Ubuntu Linux 7.04 or later, x86-32
- Macintosh
    - OS X
    - OS X 10.2.6 or higher with Java 1.4 installed
    - OS X 10.4.1 or later
- Multi-computer systems
    - Meiko and BBN
    - Sun3, Sun 4, and HP 9000 workstations
- OS/2
- OS/370
- OS/400
- SGI
- Sun Solaris
- Sparc/Intel Solaris
- SunOS
- Unix
- Windows
    - Windows 3.1
    - Windows 95
    - Windows 98
    - Windows NT
    - Windows 2000
    - Windows XP
    - Windows as a DOS Application
    - Windows Vista, x86-32

5.19

As depicted, the majority of toolkits run on Windows and Linux, although there is a large contingent that runs on Macintosh. There also is a growing trend toward implementing and running models in Java, both because of the simplicity of programming and also because of the platform independence that Java offers. We also can see this trend in table 6 (Toolkits That Run on Various Windows Operating Systems). An important note for the reader is that we tried to look at each toolkit as completely and comprehensively as possible. We gathered this information from open source documentation provided by the authors and by third parties who used the platform. However, this table is not complete; Rather, it is a baseline of platforms that have been known and documented to work on particular operating systems. Note that this does not necessarily exclude toolkits from running on additional operating systems. For example, a toolkit that runs on Windows NT may also run on Windows 2000, Windows XP, and Windows Vista. In places where this table is not complete, it is because the developers have not specified the complete granularity of the platforms with respect to different operating systems.[2]

Platforms Per Subcategory

5.20

We will begin with the Windows platforms. Where specified in documentation, we have decomposed the platforms into Windows 3.1, 95, 98, 2000, ME, NT, XP, and Vista. See table 6. Again, the reader should note that this is not a complete representation; rather it presents a categorization of platforms based on *documented* success for each platform on each operating system.

Table 6: Toolkits That Run On Various Windows Operating Systems

Operating System	Toolkit
Windows (version not specified)	AgentSheets Ascape Breve Cormas DOMAR LSD Matlab Moduleco StarLogo StarLogo TNG OBEUS oRIS PS-I Repast.Net SeSAm SimAgent (without graphics) Swarm
Windows 3.1	SDML
Windows 95	ABLE iGen SDML

	Zeus
Windows 98	ABLE Cougaar iGen SDML Zeus
Windows NT	ABLE AgentBuilder Cougaar iGen MIMOSE (Java based client) SDML Zeus
Windows 2000	AgentBuilder ADK Brahms iGen SDML Zeus
Windows XP	AgentBuilder ADK AnyLogic Brahms Cougaar iGen
Windows as a DOS Application	MOOSE SimPlusPlus
Windows Vista, x86-32	AnyLogic

5.21
The next prominent platform is Linux and its distributions such as Ubuntu and SuSE. These are depicted in table 7.

Table 7: Toolkits That Run On Linux Operating Systems

Operating System	Toolkit
Linux (version not specified)	AgentBuilder ADK Ascape Brahms Breve Cormas Cougaar DOMAR MAGSY MAML Matlab MIMOSE (client/server version) MIMOSE (Java based client) Moduleco MOOSE oRIS SimAgent StarLogo PS-I SDML SeSAm Swarm Zeus
X86 or x86_64 linux	DeX
SuSE Open Linux 10.2 or later, x86-32	AnyLogic
Ubuntu Linux 7.04 or later, x86-32	AnyLogic

5.22
Next we have toolkits supported by the Macintosh operating system (table 8). These include toolkits such as Ascape, AgentSheets, Cormas, Cougaar, Brahms, Breve, SeSAm, StarLogoTNG, and Swarm.

Table 8: Toolkits That Run On Macintosh Operating Systems

Operating System	Toolkit
Macintosh (version not specified)	Ascape Cormas LSD Matlab

	Moduleco MacStarLogo StartLogo T StarLogo TNG
OS X	AgentSheets Brahms Breve Cougaar SeSAm Swarm
OS X 10.2.6 or higher with Java 1.4 installed	StarLogo
OS X 10.4.1 or later	AnyLogic

5.23

The next major contingent of toolkits are those that will run on any machine that has a Java Virtual Machine (JVM) or Java Runtime Environment (JRE) installed. These are depicted in table 9. Where specified in the documentation, we have decomposed these into several subcategories, including any platform with a Java Virtual Machine, at least Software Development Kit (SDK) 1.4.1 or later, SDK 2.0 or later, Java Runtime Environment 1.5 or later, and several more. Again, the reader should note that this is not a complete representation; rather it presents a categorization of platforms based on *documented* success for each platform on each operating system; Thus, some toolkits may work with additional virtual machines or subcategories.

Table 9: Toolkits That Run on Various Java Virtual Machines

Type of Java Virtual Machine	Toolkit
Any platform with a Java Virtual Machine (JVM)	AgentBuilder ADK AgentSheets jECHO ECJ Jade JAS JASA JCA-Sim jES JESS RepastS RespastJ
Java 2 JVM	ABLE Madkit
SDK version 1.4.1 or later	ADK NetLogo SimPack
Java 2 SDK	Sugarscape
Java Runtime Environment (JRE) 1.5.0 or later	AnyLogic JAS
Java SDK 5.0 or better	ADK (but must contact for support) SeSAm
JDK 1.1	VSEit (Java 1.1.7 or later)
Any Java Development Kit (JDK) installation	FAMOJA Omonia
Java 2 Runtime Environment(JRE) and Internet Explorer 5.x or greater	Sugarscape
JRE Java version 1.4	ADK
JRE version 1.3.1	ADK Mason
Bea's JRockit JVM	ADK
IBM's JVM	ADK

5.24

Another prominent platform is Unix. Toolkits that will run specifically on Unix platforms are:

- ABLE
- ADK
- Ascape
- Cormas
- DOMAR
- ECHO
- LSD
- MAGSY
- StarLogo
- SDML

- SimAgent
- SME
- Zeus

5.25 Finally, we also have a small contingent of toolkits that run on the Sun Solaris platform.

- AgentBuilder
- ADK
- MAGSY
- Matlab
- MIMOSE (client/server version)
- MIMOSE (Java based client)
- SDML
- Zeus

5.26 Table 10 depicts the remaining domain of toolkits per operating system.

Table 10: Remaining Toolkits per Operating System

Operating System	**Toolkit**
ADUX	SDML
AIX	SDML
Any platform that supports C++/any C++ compiler	SimBioSys SimPack SimPlusPlus
BSD	Zeus
DOS	MOOSE SimPlusPlus
Emulation of Windows NT or Linux	PS-I
HPUX	SDML
IA32 Linux; PPC Linux	oRIS
IBM mainframes	ADK (paid support)
Java-1.4-capable PDAs	Cougaar
Multi-computer systems Meiko and BBN Sun3, Sun 4, and HP 9000 workstations	Jade's sim++
OS/2	ABLE SimPlusPlus (as a DOS application)
OS/370 OS/400	ADK
SGI	MOOSE oRIS SDML
Sparc/Intel Solaris	Brahms (available upon request)
SunOS	MAGSY MOOSE

Domain

5.27 In this section, we examine various domains for which the toolkits are specialized. Many of the toolkits are specifically tailored for particular domains, and many are general purpose toolkits that can be used for a variety of domains. The main domains are shown below.

- Applied simulations/electronic CAD
- Artificial Intelligence (general purpose)
  - Machine learning and reasoning
  - Social sciences
  - 3D simulations
    - multi-agent systems
  - Human-like intelligent agents
- Biology
- Cellular automata
- Computational economics/Auction mechanisms
- Distributed simulations
  - Large scale
  - Mobile (distributed) agents
  - Small scale embedded
  - Virtual reality
  - Highly distributed, scalable, reliable, survivable applications
- Ecological modeling
- Education

    - Artificial Intelligence
    - For students to model the behavior of decentralized systems
    - Implementing software agents
    - Learning (including explanation based learning)
    - Teaching computer simulation
        - At the undergraduate (senior) and graduate levels
        - K-12 social sciences, social studies, math, and science
    - Using object oriented principles
    - Teaching programming techniques to students new to simulation
    - Scientific and engineering math and computation; data analysis, exploration, and visualization
- Enterprises
    - Single enterprise
    - Systems of enterprises
- Evolutionary computing
- General purpose agent based
    - Multi-agent systems with agent based simulation layer
    - General-purpose parallel applications
- Human performance modeling
    - Training systems
    - Performance support systems
- Multi-agent systems (general purpose)
    - Large scale distributed
    - Complex environments
    - Distributed
    - Decision-making in complex environments (with limited rationality)
    - Organizational processes
- Natural resources management
- Natural Sciences
- Political phenomena
- Rule engine and scripting environment
- Simulating organizational processes
- Social sciences
    - Help beginning users get started authoring models
    - Social systems
- Testing Base24 applications
- Urban simulation

5.28

The major specializations are agent based systems, artificial intelligence, distributed simulation, education, multi-agent systems, and social and natural sciences. An important note for the reader is that these are the *primary* domains for which the toolkit has been designed, and these are the primary domains for which the toolkit has been documented as a primary domain. Note that many toolkits are used for more domains that just their primary. However, the secondary domains have extremely unequal and incomplete representation. Therefore, we do not attempt to classify toolkits further than their primary domain. Also note that the domain categories listed here are the terminology of the toolkit documentation. As such, we do not attempt to disambiguate domain terminology. Rather, the goal is to give the user a broad feel for the types of domains for which these toolkits may be applicable, so that it will bring to the fore potential toolkits that the user otherwise may not have considered. The user should then explore further the differences between similar terminology in the domain categories of interest.

Platforms Per Subcategory

5.29

We will begin with general purpose agent based platforms (table 11). These toolkits are not geared toward special domains but rather can be used for general classes of agent based simulation. These include toolkits such as Swarm, Mason, Magsy, AgentBuilder Lite/Pro, Anylogic, Madkit, DeX, DOMAR, and Ascape. One toolkit of note is Madkit. It actually is a multi-agent platform, but it includes an agent based simulation layer. Another toolkit of note is DeX. DeX has an additional special emphasis on parallel applications.

Table 11: General Purpose Agent Based Toolkit

Purpose	Toollkit
General purpose agent based	AnyLogic Ascape DOMAR JAS MAGSY Mason SeSAm SimPack Swarm
Multi-agent systems with agent based simulation layer	Madkit
General-purpose parallel applications	DeX

5.30

Next, we have toolkits that specialize in distributed simulation (table 12). Here we see several toolkits that have even more particular specializations within this domain. For example, Cougaar and Tryllian Agent Development Kit specialize in large scale distributed applications, whereas oRIS specializes in virtual reality. Cougaar, in addition, has subspecializations in scalable, reliable, survivable and small scale embedded applications.

Table 12: Toolkits Specializing in Distributed Simulation

Type of Distributed Simulation	Toolkit
General purpose distributed simulations	AnyLogic DOMAR Jade Madkit
Large scale	ADK Cougaar SimAgent
Mobile (distributed) agents	ADK
Small scale embedded	Cougaar
Virtual reality	oRIS
Highly distributed, scalable, reliable, survivable applications	Cougaar

5.31 Another main focus is education. Few toolkits are oriented toward education as their primary specialization. The forerunners as pedagogical tools are AgentSheets, StarLogoT, NetLogo, oRIS, and StarLogo (and decedents OpenStarLogo and StarLogoTNG). Since there is a strong interest in education (Serenko 2002), and because many of the platforms can be and are being used for pedagogical purposes in addition to their primary specialization, we have included in our educational taxonomy toolkits with secondary educational foci (see table 13). There are general purpose educational platforms, and there are toolkits that specialize in particular aspects of education. The general purpose toolkits include StarLogo, NetLogo, StarLogoT and MIMOSE. Within educational subspecializaties, there are toolkits that specialize in teaching programming techniques (Matlab, Sugarscape), object oriented principles (jECHO), math and computation (Matlab), how to model decentralized systems (StarLogo, StarLogoT, StarLogoTNG), computer simulation (FAMOJA, oRIS, Matlab, SeSAm), and implementing software agents (Brahms, SimAgent). Teaching computer simulation is further specialized for K-12 students (AgentSheets, StarLogo), and undergraduate (senior) and graduate level students (SimPack).

Table 13: Documented Platforms With a Primary or Secondary Pedagogical Focus

Pedagogical Focus	Toolkit
General purpose education[3]	MIMOSE NetLogo StarLogo StarLogoT StarLogo TNG VSEit
Artificial Intelligence	Breve
For students to model the behavior of decentralized systems	StarLogo StarLogoT StarLogo TNG
Implementing software agents	Brahms SimAgent
Learning (including explanation based learning)	SOAR
Teaching computer simulation	FAMOJA Matlab oRIS SeSAm
At the undergraduate (senior) and graduate levels	SimPack
K-12 social sciences, social studies, math, and science	AgentSheets StarLogo
Using object oriented principles	jECHO
teaching programming techniques to students new to simulation	Matlab NetLogo StarLogoTNG Sugarscape
scientific and engineering math and computation; data analysis, exploration, and visualization	Matlab

[3] Few toolkits are oriented toward education as their primary specialization. The forerunners as pedagogical tools are AgentSheets, StarLogoT, NetLogo, oRIS, and StarLogo (and decedents OpenStarLogo

and StarLogoTNG). Since there is a strong interest in education (Serenko 2002), and because many of the platforms can be and are being used for pedagogical purposes in addition to their primary specialization, we have included in our educational taxonomy toolkits with secondary educational foci.

5.32

The next major specialization is multi-agent systems (table 14). While many of these toolkits do have support and are being used for agent based modeling, their main purpose is for building multi-agent systems. These toolkits include Brahms, Cormas, Cougaar, Jade, Madkit, Magsy, Moduleco, oRIS, and SDML.

Table 14: Toolkits With Primary Specialization in Multi-Agent Systems

Type of Multi-agent System	Toolkit
Multi-agent systems (general purpose)	AgentBuilder oRIS MAGSY
Large scale distributed	ADK Cougaar
Complex environments	SimAgent
Distributed	Jade
Decision-making in complex environments (with limited rationality)	SDML
Organizational processes	Brahms

5.33

The next major contingent of toolkits are those that specialize in artificial intelligence. These include toolkits that are geared for artificial intelligence in general, for machine learning, for creating human-like intelligent agents, and for artificial intelligence for the social sciences in particular (See table 15).

Table 15: Toolkits With a Primary Specialization in Artificial Intelligence

Type of Artificial Intelligence Focus	Toolkit
Artificial Intelligence (general purpose)	Breve iGen SOAR
Machine learning and reasoning	ABLE Zeus
Social sciences	Omonia
3D simulations multi-agent systems	Breve
Human-like intelligent agents	SimAgent

5.34

In table 16, we see the toolkits that are geared specifically for the social sciences. Again, there are general purpose toolkits as well as particular subspecializations within.

Table 16: Toolkits With A Primary Specialization Toward the Social Sciences

Type of Social Science Specialization	Toolkit
General purpose Social Sciences	AgentSheets LSD FAMOJA MAML MAS-SOC MIMOSE NetLogo Repast SimBioSys StarLogo StarLogoT StarLogo TNG Sugarscape VSEit
Help beginning users get started authoring models	NetLogo
Social systems	Moduleco

5.35

Finally, we have toolkits that are geared toward the natural sciences. These are:

- Matlab
- NetLogo
- OpenStarLogo
- StarLogo
- StarLogoT
- StarLogo TNG

5.36
The rest of the domains are highly specialized and only have one or two supporting toolkits. These can be found in the table 17.

Table 17: Highly Specialized Documented Primary Domains Across the ABM Spectrum

Documented Primary Domain	**Toolkit**
Applied Simulations/Electronic CAD	Jade's Sim++
Biology	SimBioSys
Cellular automata	JCA-Sim
Computational economics/Auction mechanisms	JASA
Ecological modeling	ECHO jEcho SME
Enterprises	jES
Evolutionary computing	ECJ
Human performance modeling	iGen SOAR
Training systems	iGen
Performance support systems	iGen
Natural Resources Management	Cormas
Political phenomena	PS-I
Rule engine and scripting environment	JESS Zeus
Simulating organizational processes	Brahms
Testing Base24 applications	SimPlusPlus
Urban simulation	OBEUS

User Support

5.37
Another important category that individuals and organizations look at when determining a toolkit to use is the degree of support that is available to the user. In this section we examine the types of user support that are available. These include project wikis, documentation (such as user manuals), consulting, bug lists, formal training, example models, tutorials, third party extensions, selected references, application programming interfaces (APIs), and frequently asked questions (FAQ) sections. The types of support in the user support domain are:

- API
- Bug List
- Consulting
- Documentation
- Example Models
- FAQ
- Formal Training
- Selected References
- Tutorials
- Third Part Extensions
- Wiki

Platforms Per Subcategory

5.38
Next, we have toolkits that have some form of user documentation. By user documentation, we are looking for manuals that explain how to use the toolkit. Almost every platform comes with a user manual. Note that in this survey, we do not attempt to compare the comprehensiveness of the user manuals; we merely are mentioning that the toolkit at least comes with some degree of documentation. We did, however, note several toolkits in particular that have limited documentation (as indicated below).

- ABLE
- ADK
- AnyLogic
- Ascape
- Brahms
- Breve

- Cormas
- Cougaar
- DeX
- ECJ
- FAMOJA
- iGEN
- JADE
- JAS
- JASA (limited)
- JCA-Sim
- jES (limited)
- JESS
- LSD
- MacStarLogo
- Magsy (limited)
- Mason
- Matlab
- MIMOSE
- Moduleco (limited)
- MOOSE
- NetLogo
- OBEUS
- Omonia
- OpenStarLogo
- oRIS
- PS-I
- Repast
- SDML (limited)
- SimAgent
- SimPack
- SME
- SOAR
- StarLogo
- StarLogoT
- StarLogo TNG
- Swarm
- VSEit
- Zeus

5.39

Some toolkits have tutorials set up to assist the user in getting started authoring models. These toolkits are:

- ABLE
- AgentSheets
- Brahms
- Breve
- Cougaar
- ECJ
- FAMAJO
- JADE
- JAS
- MAML
- Mason
- NetLogo
- Repast
- SDML
- SeSAm
- SimAgent
- SOAR
- StarLogo
- StarLogoT
- StarLogo TNG
- Swarm

5.40

Again, in this survey, we do not attempt to evaluate the quality of the tutorials, only that these toolkits have tutorials established to support the user.

5.41

Third, we identify toolkits which include mailing lists, listservs, or online forums to support the user:

- ABLE
- ADK
- AgentBuilder
- AnyLogic
- Ascape
- Brahms
- Breve
- Cormas
- Cougaar

- DeX
- ECJ
- JADE
- JASA
- JESS
- Madkit
- Mason
- NetLogo
- OpenStarLogo
- Repast
- SDML
- SeSAm
- SimPack
- SME
- SOAR
- StarLogo
- StarLogo TNG
- Swarm

5.42

Many toolkits also have a section for frequently asked questions. These toolkits include:

- ABLE
- ADK
- AgentBuilder
- AgentSheets
- Breve
- Cougaar
- JADE
- JESS
- Madkit
- NetLogo
- OpenStarLogo
- Repast
- SeSAm
- SOAR
- StarLogo
- StarLogo TNG
- Swarm

5.43

Some toolkits include their APIs. These are depicted below.

- ABLE
- ADK
- Ascape
- Breve
- DeX
- DOMAR
- ECJ
- FAMOJA
- JADE
- JAS
- JASA
- JCA-Sim
- Mason
- Moduleco
- oRIS
- StarLogoT
- StarLogo TNG
- Sugarscape

5.44

Some toolkits also have defect (also known as "bug") lists.

- ADK
- AgentBuilder
- Breve
- JADE
- Madkit
- Matlab
- NetLogo
- OpenStarLogo
- Repast
- SOAR
- StarLogo
- StarLogoT
- Swarm
- VSEit

5.45

In addition to tutorials, many toolkits also have example models available.

- ADK
- Cormas
- ECJ
- JCA-Sim
- JESS
- LSD
- Madkit
- MAML
- Omonia
- OpenStarLogo
- oRIS
- Repast
- SimAgent
- SOAR
- Swarm
- VSEit

5.46

These differ from tutorials because these are more comprehensive templates that individuals can use to help them use author models. They may not necessarily come with directions on how to use the toolkit in general. Tutorials, on the other hand, are designed to walk the user step by step through how to use the toolkit. They may or may not include model templates.

5.47

Some toolkits also include selected references/publications that users can read for more information on the toolkit. These are depicted below.

- ABLE
- AgentSheets
- AnyLogic
- Ascape
- Cormas
- Cougaar
- ECHO
- iGEN
- JASA
- MAML
- Mason
- MOOSE
- NetLogo
- PS-I
- Repast
- SDML
- SimAgent
- SimPack
- SOAR
- Swarm

5.48

Several of the toolkits have consulting services available in conjunction with their toolkit.

- AnyLogic
- AgentBuilder
- iGen
- Matlab

Note that these are all proprietary toolkits, so the user probably will have to pay extra for these services.

5.49

Some toolkits also have specialized training classes available for users.

- AgentBuilder
- AgentSheets
- AnyLogic
- Cormas
- iGEN
- Matlab

Again, these are mostly proprietary toolkits.

5.50

Some toolkits have links on their website to third party extensions that individuals have developed to fulfill a specialized need.

- JESS
- Mason
- Matlab
- NetLogo
- Repast
- SOAR

For example, Mason has third party extensions that aid in social network statistics, rigid body physics, and integration with the Jung social network system (Mason website).

5.51

Finally, we have toolkits that have project wikis as part of user support (shown below).

- Brahms
- FAMOJA
- JESS
- SeSAm
- SOAR
- Swarm

Characteristics per platform

5.52

In this section, we compiled all of the preceding information into an easy-to-use matrix. Whereas the previous section aids the user in viewing toolkits across one or two characteristics, this section is helpful for the user who wants to examine all of the characteristics across one platform. For better viewing purposes, we split the matrix into two submatrices (See Appendices 2 and 3).

Conclusion: On-going and Future Work

6.1

We have developed a web-based tool that incorporates all of our findings so far. This is a searchable repository of ABM platforms into which users input a range of characteristics, and the tool returns a list of candidate platforms that contain those characteristics. This tool is available at http://agent.cse.nd.edu/abmsearchengine.php.

6.2

We also have created a corresponding page in Wikipedia based on this research. In addition to summarizing our current results, we include several categories of interest concerning 3D and GIS capabilities. The article is entitled "ABM Software Comparison," and it is linked from the main "Agent Based Model" wiki. We invite the ABM community to participate in expanding this research further.

6.3

In the future, we would like to continue our research into various ABM toolkits. In particular, we would like to examine more complex characteristics across the ABM spectrum. In our next work, we will design a survey to explore characteristics such as ease of use, degree of programming required, maximum number of agents supported, statistical support, and feature completeness.

6.4

In this paper, we have begun a comprehensive survey of ABM platforms. We gathered as many platforms as possible that were being used for ABM purposes, and we began to classify them. In particular, we examined 5 characteristics in depth: programming language required, type of license governing the toolkit, type of operating system required, primary domain for which the toolkit has been designed, and degree of support available to the user. Our goal was to give project designers the capability to easily compare toolkits based on these characteristics and to help him/her better choose a toolkit that suits his/her needs. As such, we have included a range of general as well as specialized toolkits. Some of the toolkits have never been included in surveys before, and we hope that including these will help individuals choose toolkits that are more suited for their projects rather than having to "redesign the wheel." In order to facilitate comparison, we created several taxonomies which have been presented here in tabular form. With these representations, the user can quickly examine one characteristic across a range of toolkits as well as a range of characteristics across one toolkit.

Acknowledgements

We would like to thank Curtis Blais, Ranjeev Mittu, and Amy Henninger for their assistance in this research.

Notes

[1] There is some ambiguity concerning the terms "platform" and "toolkit." From the Computer Science domain, the term "toolkit" denotes the application level software package, and the tern "platform" denotes the underlying hardware on which the software runs. In the Social Science domain, on the other hand, the term "platform" and "toolkit" have been used interchangeably (Gilbert 2002; Tobias 2004). In this paper, we also use the terms "platform" and "toolkit" interchangeably.

[2] Please note this limitation in this section and throughout the remainder of the paper.

Appendix 1: Glossary of Acronyms

ABLE
Agent Building and Learning Environment

ADK
Tryllian Agent Development Kit
API
Application Programming Interface
BSD
Berkley Software Distribution
Cormas
Common-pool Resources and Multi-Agent Systems
DOMAR
Distributed operator model architecture
FAMOJA
Framework for Agent-based Modelling with Java
FAQ
Frequently Asked Questions
GPL
GNU General Public License
JASA
Java Auction Simulator API
JDK
Java Development Kit
jES
Java Enterprise Simulator
JRE
Java Runtime Environment
JVM
Java Virtual Machine
LGPL
GNU Lesser General Public License
LSD
Laboratory for Simulation Development
MAGSY
Rules Based Multi-Agent System
Matlab
Matrix Laboratory
MIMOSE
Micro-und Multilevel Modelling Software
Madkit
Multi Agent Development Kit
MAML
Multi-agent modeling language
MAS-SOC
Multi-Agent Simulations for the SOCial Sciences
MOOSE
Multimodeling Object-Oriented Simulation Environment
OBEUS
Object Based Environment for Urban Simulation
PS-I
Political Science- Identity
Repast
REcursive Porous Agent Simulation Toolkit
SDK
Software Development Kit
SeSAM
Shell for Simulated Agent Systems
SME
Spatial Modeling Environment
SDML
Strictly Declarative Modeling Language
VSEit
Versatile Simulation Environment for the Internet

Appendix 2: Characteristics Per Platform - Domain, License, and Programming Language Required

Platform	Domain[1]	License	Programming Language Required
Agent Building and Learning Environment (ABLE)	Building intelligent agents using machine learning and reasoning	Open source (free for academic and non-commercial use)	Able Rule Language (ARL)
AgentBuilder Lite/Pro	General purpose multi-agent systems	Proprietary; Discounted academic licenses available	Knowledge Query and Manipulation Language (KQML); Java; C; C++[2]
Tryllian Agent Development Kit (ADK)	Large scale distributed applications; Mobile (distributed) agents	Dual licensed: either accept the LGPL or contact	Java

		Tryllian to acquire a closed source license	
AgentSheets	Teaching simulation to grades K-12 in social studies, mathematics, sciences, and social sciences	Proprietary	Visual AgenTalk (VAT); a rule-based visual programming language; can be exported to Java; programming by example and programming by demonstration
AnyLogic	Agent based general purpose; distributed simulations	Proprietary	Java; UML-RT (UML for real time)
Ascape	General-purpose agent-based models.	BSD	Java
Brahms	Multi-agent environment for simulating organizational processes	Free, but only available for research or non-commercial purposes	Brahms language (an agent oriented language)
Breve	Building 3D simulations of multi-agent systems and artificial life.	GPL	Simple Interpreted object oriented language called Steve; agent behaviors can be written in python
Common-pool Resources and Multi-Agent Systems (Cormas)	Natural resources management	Free to modify but not to distribute the modified version	Smalltalk (requires VisualWorks to run)
Cougaar	Multi-agent systems; highly distributed, scalable, reliable, survivable applications; Domain independent; large scale distributed, complex, data intensive (can be configured for small-scaled embedded Applications	Cougaar Open Source License (COSL) is a modified version of the OSI approved BSD License	Java
DeX	Developing, analyzing, and visualizing dynamic agent-based and multi-body simulations; parallel applications	Free (open source) - read license	C++; dML (deX Modeling Language): a domain-specific language based on C++; python
Distributed operator model architecture (DOMAR)	General purpose simulation environment	Free (open source) - read license	Java (OMAR-J); lisp (OMAR-L)
ECHO	Ecological modeling	Free, open source	C
jEcho	Ecological modeling using object oriented principles	Free, open source	Java
ECJ	Evolutionary computation; genetic programming	Academic Free License - open source	Java
Framework for Agent-based Modelling with Java (FAMOJA)	Resource flow management, theoretical systems science, applied systems, environmental systems analysis	LGPL	Java
iGen	Artificial intelligence engine; human performance modeling; embeddable cognitive agents	Proprietary (various prices for Developer's License; Modeler's License; Runtime License; and Academic Licenses)	COGNET Execution Language (CEL); C++; C; Java
JADE	Distributed applications composed of autonomous entities	LGPL version 2	Java
JAS	General purpose agent based	LGPL; associated third party licenses (usually nonÂproprietary)	Java

Java Auction Simulator API (JASA)	Computational economics; Agent based computational economics	GPL	Java
JCA-Sim	Cellular automata; General purpose simulator	Free (closed source)	Java; Cellular Description Language (CDL) (for input to simulation)
Java Enterprise Simulator (jES)	A single enterprise or a system of enterprises	Academic free license	Java
JESS	Rule engine and scripting environment	Proprietary; free for academic use	Java/Jess/JessML (declarative xml rule language)
Laboratory for Simulation Development (LSD)	A language for simulation models; social sciences	GPL	C++; LSD
Multi Agent Development Kit (Madkit)	A generic, highly customizable and scalable platform; general purpose multi-agent platform with agent based simulation layer	LGPL for basic libraries; GPL for development and non-commercial applications	Java; MadKit may be developed in all languages that are compiled into Java; for the moment, MadKit comes with 4 scripting languages which are executed in the Java Virtual Machine: Scheme (Kawa), Jess (rule based language), BeanShell (Java interpreted) and Python (jython). Using the JNI (Java Native Interface) technique, it should be possible to develop agents written in C or C++. It is also possible to embed Java agents in C/C++ applications using the same technique, using JNI as a glue between the two worlds.
Rules Based Multi-Agent System (MAGSY)	General purpose multi-agent systems	Free (closed source)	Magsy (production language)
Multi-agent modeling language (MAML)	Social science; domain specific programming language for developing agent based models	The compiler is freely downloadable for evaluation purposes (open source) Later the system will be put under GNU license	MAML language; C; visual programming interface
Mason	General purpose; social complexity, physical modeling, abstract modeling, AI/machine learning	Academic Free License (open source)	Java
Multi-Agent Simulations for the SOCial Sciences (MAS-SOC)	Social simulation	Contact authors for availability	AgentSpeak(XL), an extension of Agentspeak(L) and (Environment Description Language for Multi-Agent Simulation) ELMS, a language for modelling environments where cognitive agents are situated. Future work to implement in Java
Matrix Laboratory (Matlab)	Teaching simulation programming techniques to students new to simulation; scientific and engineering math and computation; data analysis, exploration, and visualization	Proprietary	MATLAB® is a high-level language that includes matrix-based data structures, its own internal data types, an extensive catalog of functions, an environment in which to develop your own functions and scripts,

			the ability to import and export to many types of data files, object-oriented programming capabilities, and interfaces to external technologies such as COM, Java, programs written in C and Fortran, and serial port devices.
Micro- und Multilevel Modelling Software (MIMOSE)	Social sciences; education	Free (closed source)	A model description language (derived from functional language paradigms)
Moduleco	Multi-agent platform	GPL	Java
StarLogo	Social and natural sciences; Educators; for students to model the behavior of decentralized systems; user friendly for K-12 students	Free (closed source) - Clearthought Software License, Version 1.0	StarLogo (an extension of Logo)
MacStarLogo	Social and natural sciences; Educators; for students to model the behavior of decentralized systems; user friendly for K-12 students	Free (closed source)	MacStarLogo
OpenStarLogo	Social and natural sciences; Educators; for students to model the behavior of decentralized systems; user friendly for K-12 students	Free for use and distribution for non-commercial purposes (open source)	StarLogo (an extension of Logo)
StarLogoT	Social sciences; Education; decentralized networks	Free (closed source)	StarLogoT
StarLogo TNG (The Next Generation)	Social and natural sciences; teaching basic computer programming skills	StarLogo TNG License v1.0 - (closed source) - the code may be freed up eventually. The original StarLogo is apparently going to be released under an open source license soon	StarLogo TNG language - a graphical programming language and a 3d world
NetLogo	Social and natural sciences; Help beginning users get started authoring models	Free, not open source; A quick summary of the license is that use is unrestricted, including commercial use, but there are some restrictions on redistribution and/or modification (unless you contact Uri Wilensky to arrange different terms)	NetLogo
Object Based Environment for Urban Simulation (OBEUS)	Urban simulation	Free (closed source)	Microsoft.net .NET languages - C#, C++, or Visual Basic.
oRIS	Teaching; programming by concurrent objects, multi-agent systems, distributed virtual reality, adaptive control	Proprietary - (free for academic institutions)	Oris language; Very close to C++ and Java (dynamic and interpreted multi-agent language)
Political Science-Identity (PS-I)	Political phenomena	GPL	No programming required; TCL/TK scripting to apply effects

Quicksilver (now called omonia)	AI/social sciences	LGPL	Java
Recursive Porous Agent Simulation Toolkit (Repast)	Social sciences	BSD	Java (RepastS, RepastJ); Python (RepastPy); Visual Basic, .Net, C++, J#, C# (Repast.net)
Strictly Declarative Modeling Language (SDML)	Multi-agent systems (with limited rationality)	GPL; third party license (for VisualWorks)	Smalltalk release 5i.2 Non-Commercial
Jade's sim++	Parallel simulation; Applied simulations; network planning; electronic CAD; real time communication simulation	GPL version 2	C++
SimPlusPlus	Testing Base24 applications	GPL	Fully programmable with any language that can support activeX components (e.g. C, C++, VB, VBA, Java, and others), but no programming required
SimAgent (aka sim_agent)	Research and teaching related to the development of interacting agents in environments of various degrees and kinds of complexity; exploratory research on human-like intelligent agents; systems involving large numbers of highly distributed fairly homogeneous relatively 'small' agents; primarily designed to support design and implementation of very complex agents, each composed of very different interacting components (like a human mind) where the whole thing is embedded in an environment that could be a mixture of physical objects and other agents of many sorts	Free (open source); MIT/XFREE86 license (for poplog libraries); may later be replaced by GPL	Pop-11, like Common Lisp, is a powerful extendable multi-purpose programming language supporting multiple paradigms. Within the Poplog environment Pop-11 also supports programs written in Prolog, Common Lisp or Standard ML
SimBioSys	Agent-based evolutionary simulations in both biology and the social sciences	Artistic License Agreement	C++
Multimodeling Object-Oriented Simulation Environment (Moose)	General purpose, agent abased (modeled from SimPack)	Unable to verify that available for public use	C++
SimPack	General purpose, agent based; teaching computer simulation at the underÂgraduate (senior) and graduate levels	GPL	C++; (C libraries no longer maintained); Java
Spatial Modeling Environment (SME)	Ecological economic; Ecoystems modeling	GPL	No knowledge of computer programming required
Shell for Simulated Agent Systems (SeSAm)	General purpose (agent based); teaching	LGPL	SeSAm-Impl and SeSAm-UML; Visual programming
SOAR	General purpose AI; human performance modeling; learning (including explanation-based learning)	BSD	Soar 1 to 5 in Lisp; Soar 6 in C; Java, C++, TCL
Sugarscape	Social sciences; education	GPL	Java
Swarm	General purpose agent based	GPL	Java; Objective C
VSEit	Social sciences; education	Free (closed source)	Java
ZEUS	Rules engine and scripting environment; Distributed multi-agent simulations	Open source (read license)	Visual editors and code generators

[1] An important note for the reader is that these are the primary domains for which the toolkit has been designed. Note that many toolkits are used for more domains that just their primary domain. However, the secondary domains have extremely unequal and incomplete representation. Therefore, we do not attempt to classify toolkits further than their primary domain.

[2] Developer-defined interagent communications commands; built-in Java classes (supplied by the AgentBuilder toolkit) and domain-specific Java classes provided by the developer. All of these classes used by AgentBuilder agents are referred to as Project Accessory Classes (PACs); PACs can be written entirely in Java, or can be written in C/C++ and invoked via the Java Native Interface (JNI)

Appendix 3: Characteristics Per Platform - Platform Supports, User Support, and Website

Platform	Platform Supports[3]	User Support	Website
Agent Building and Learning Environment (ABLE)	OS/2; Windows 95; Windows 98; Windows NT; and UNIX (any Java 2 JVM)	FAQ; tutorial; examples; discussion forum; emailing developers; selected publications; API; documentation	http://www.alphaworks.ibm.com/tech/able
AgentBuilder Lite/Pro	Windows NT; Windows 2000; Windows XP; Linux; Sun Solaris; any platform with a Java Virtual Machine	Consulting; training; example; FAQ; users manuals; defect reporting; mailing list	http://www.agentbuilder.com/Documentation/Lite/
Tryllian Agent Development Kit (ADK)	Windows; Unix; Big Iron IBM mainframes[4]; anywhere that the Java Standard Edition version 1.4 runs; Sun Java Runtime Environment version 1.3.1 or 1.4; JDK 5.0[5]; any platform on which Sun has made available a JVM; Windows 2000; Windows XP; Solaris; GNU/Linux; Additionally, the ADK has been tested on OS/400 and OS/370. Apple's OS X is not supported, but part of the development of the ADK is done on OS X.	FAQ; defect reporting; documentation; mailing list; quickstart guide; examples; email maintainer for more support; API	http://www.tryllian.com
AgentSheets	Windows; Macintosh OS X (PowerPCs) Macintosh OS X (Intel Macs); should run on any Java Virtual Machine	Manuals; tutorial movies; FAQ; recommended readings on programming and simulation; personal contact with developers; elementary school training; teacher guides	http://www.agentsheets.com/index.html

AnyLogic	AnyLogic 6 models are standalone Java applications (or applets) and run on any Java-enabled platform or in any Java-enabled browser with the following version of JRE (Java Runtime Environment):JRE 1.5.0 or later; Java plug-in (needed to run models in a Browser) is optionally installed with the JRE; Windows Vista, x86-32; Windows XP, x86-32; Apple Macintosh OS X 10.4.1 or later, Universal; SuSE Open Linux 10.2 or later, x86-32; Ubuntu Linux 7.04 or later, x86-32	Demos; training; consulting; knowledge base; online forum; ask a question; documentation; selected references	http://www.xjtek.com/anylogic/
Ascape	Windows; Macintosh; Unix; Linux; web	Online forum (emailing list); selected references; documentation; API	http://ascape.sourceforge.net/index.html#Introduction
Brahms	Windows 2000; Windows XP; Linux; Sparc/Intel Solaris; and Macintosh OS X	Documentation; API; tutorials; discussion forums; email contacts	http://www.agentisolutions.com/index.htm
Breve	Macintosh OS X; Linux; and Windows	Email developer; tutorials; FAQ; forums; defects section; API; documentation	http://www.spiderland.org/node/2602
Common-pool Resources and Multi-Agent Systems (Cormas)	Linux; Macintosh; Unix; Windows	Training, selected references; examples; online forum; email developers; documentation	http://cormas.cirad.fr/indexeng.htm
Cougaar	Windows 98; Windows NT; Windows XP; Linux; Macintosh OS X; and Java-1.4-capable PDAs	FAQ; tutorials; slide shows; documentation; selected references; email support; public forums; mailing lists	http://www.cougaar.org/
DeX	X86 or x86_64 Linux	Users guide; demo; API; peer to peer account; author support	http://dextk.org/dex/index.html
Distributed operator model architecture (DOMAR)	Windows; Unix; Linux	API; technical support from authors	http://omar.bbn.com/
ECHO	Unix workstations;	A few selected publications;	http://www.santafe.edu/~pth/echo/

	Developed on Sun Sparc architecture using Sunos 4.1.3	one outdated publication on how to compile and use Echo	
jEcho	Any Java platform	Limited documentation; Author has limited time to work with clients	http://www.brianmcindoe.com/
ECJ	Any Java platform	Tutorials; examples; API; documentation; online mailing list	http://cs.gmu.edu/~eclab/projects/ecj/
Framework for Agent-based Modelling with Java (FAMOJA)	JDK installation	Tutorial; API; wiki; documentation;	http://www.usf.uos.de/projects/famoja/
iGen	Windows 95, 98, 2000, NT, XP	Consulting; training; selected publications; (user's forum and documentation under construction, but not online yet)	http://www.cognitiveagent.com/
JADE	Any Java platform	FAQ; mailing list; defect list; tutorials; API; documentation	http://jade.tilab.com
JAS	Any Java platform version 1.5 or higher	API; documentation; tutorials; email authors	http://jaslibrary.sourceforge.net/
Java Auction Simulator API (JASA)	Any Java platform	Public forum, not very well used; API; small set of selected readings; limited documentation	http://sourceforge.net/projects/jasa/
JCA-Sim	Any Java platform	Examples; documentation; API; one contact listed	http://www.jweimar.de/jcasim/
Java Enterprise Simulator (jES)	Any Java platform	limited documentation	http://web.econ.unito.it/terna/jes/
JESS	Java Virtual Machine	FAQ; documentation; mailing list; examples; third party plug ins and libraries; wiki	http://herzberg.ca.sandia.gov/jess/
Laboratory for Simulation Development (LSD)	Windows; Unix; Macintosh	Documentation; a couple of examples; 2 contacts on webpage (but have to dig for them)	http://www.business.aau.dk/lsd/lsd.html
Multi Agent Development Kit (Madkit)	JVM (Java 2)	FAQ; documentation; online forum; examples; defect list	http://www.madkit.org/
Rules Based Multi-Agent System (MAGSY)	UNIX, LINUX, SunOS and Solaris systems.	Limited documentation; some example (inside	http://www-ags.dfki.uni-sb.de/~kuf/magsy.html

		installation package); no users support groups; no contact even for authors	
Multi-agent modeling language (MAML)	PC; Linux	Tutorial; examples; reference papers; contact developers	http://www.maml.hu/
Mason	Any Java platform (1.3 or higher)	Mailing list; documentation; Tutorials; third party extensions; reference papers; API	http://cs.gmu.edu/~eclab/projects/mason/
Multi-Agent Simulations for the SOCial Sciences (MAS-SOC)	Information not available	Information not available	http://inf.ufrgs.br/massoc (project page not available)
Matrix Laboratory (Matlab)	Windows; Linux; Solaris; Macintosh	Training; consulting; documentation; third party products and services; multiple support groups; defect reports	http://www.mathworks.com/access/helpdesk/help/techdoc/matlab_product_page.html
Micro- und Multilevel Modelling Software (MIMOSE)	Client/server version on Sun/Solaris/ and Linux; Java based client on Windows NT, Solaris, and Linux	User's manual	http://www.uni-koblenz.de/~moeh/projekte/mimose.html
Moduleco	Windows; Linux; Macintosh	API; minimal documentation	http://www.cs.manchester.ac.uk/ai/public/moduleco/
StarLogo	Macintosh OS X 10.2.6 or higher with Java 1.4 installed; Windows; Unix; Linux (StarLogo does not seem to be compatible with Java 5/1.5 on Solaris)	Mailing list; tutorials; FAQ; bug list; documentation; developer contacts	http://education.mit.edu/starlogo/
MacStarLogo	Macintosh	Download available from StarLogo webpage, but not actively developed anymore	Link removed Off the starlogo webpage
OpenStarLogo	Macintosh OS X 10.2.6 or higher with Java 1.4 installed; Windows; Unix; Linux (StarLogo does not seem to be compatible with Java 5/1.5 on Solaris)	FAQ; defects; online support lists; examples and documentation	http://education.mit.edu/openstarlogo/
StarLogoT	Macintosh	Tutorials; API; documentation; defect list; contact authors	http://ccl.northwestern.edu/cm/starlogoT/
StarLogo TNG (The Next Generation)	Macintosh and Windows	Tutorials; FAQ; documentation; mailing lists; API	http://education.mit.edu/starlogo-tng/index.htm

NetLogo	Any Java Virtual Machine, version 1.4.1 or later, is installed. Version 1.5.0_12 or later is preferred.	Documentation; FAQ; selected references; tutorials; third party extensions; defect list; mailing lists	http://ccl.northwestern.edu/netlogo/models/
Object Based Environment for Urban Simulation (OBEUS)	Windows	User's manual	http://www.enib.fr/~harrouet/oris.html
oRIS	IA32 Linux; PPC Linux; SGI Irix; and Windows	Documentation; examples in French; API	http://www.enib.fr/~harrouet/
Political Science-Identity (PS-I)	Cross platform with binaries available for win32; Windows; Linux; PS-I is not currently available for Macintosh users except via emulation of a Windows, NT, or Linux environment.	Documentation; selected publications	http://ps-i.sourceforge.net/
Quicksilver (now called omonia)	JDK installation	Examples; little documentation	http://www.xlog.ch/omonia
Recursive Porous Agent Simulation Toolkit (Repast)	Java version 1.4, although a 1.3 version for Machintosh OS X is available. To run the demonstration simulations, you'll need a Java Runtime Environment (RepastS, RepastJ); platform independent (RepastPy); Windows (Repast.net)	Documentation; mailing list; defect list; reference papers; external tools; tutorials; FAQ; examples	http://repast.sourceforge.net (RepastS) http://repast.sourceforge.net/repast_3/download.html (RepastPy, RepastJ, Repast.net)
Strictly Declarative Modeling Language (SDML)	Windows 3.1; Windows 95; Widows 98; Windows 2000; Windows NT; Linux; Intel; PowerMac; Unix; ADUX/AIX/HPUX/SGI/Solaris	Mailing list; tutorial; selected references; limited documentation included with software package	http://cfpm.org/sdml/
Jade's sim++	Available for Meiko and BBN multiÂcomputer systems and can be used on a network with Sun3, Sun 4, and HP 9000 workstations	Information not available	no longer available
SimPlusPlus	Sim++ can be used with C code or C++ code, but you MUST have a C++ compiler. DOS; Windows (as a DOS application) or OS2 (as a DOS app). The SimPack software	Author contact	http://www.simplusplus.com/

	is currently being overhauled to use C++ exclusively; however, it will still be possible to use C programs, as before, to access the C++ routines.		
SimAgent (aka sim_agent)	At least prolog version 15; Windows; Macintosh OS X; Linux; Unix	Tutorials; documentation; Selected publications; examples; author contact	http://www.cs.bham.ac.uk/research/projects/poplog/packages/simagent.html
SimBioSys	Any platform that supports C++	None	http://www.lucifer.com/~david/SimBioSys/
Multimodeling Object-Oriented Simulation Environment (Moose)	IBM PC running DOS/Windows or Version of Unix (such as Linux or BSD); Unix Workstations (SUN, SGI)	Selected references; user's manual in toolkit package	http://www.cise.ufl.edu/~fishwick/moose.html
SimPack	Any platform that supports C++; Technically, the processing environment is supposedly checked for Java 1.4 but Java 1.5 seems to work fine. Simpackj has been tested with 1.5 and exhibits no issues. The SDK is preferred over the JRE, as this could be useful for certain types of Java code that you may be writing. The SDK includes a JRE	Selected publications; mailing list; user's manual	http://www.cis.ufl.edu/~fishwick/simpack/simpack.html
Spatial Modeling Environment (SME)	Unix	Documentation; mailing list (but wasn't functional when went to the website)	http://www.uvm.edu/giee/SME3/
Shell for Simulated Agent Systems (SeSAm)	Java 5.0 or better; Windows; Linux; Macintosh OS X	Tutorials; mailing list; FAQ; wiki; author contact	http://www.simsesam.de/
SOAR	Windows 98; Windows ME; Windows 2000; Windows XP; Linux; OS X	Documentation; FAQ; selected publications; defect list; third party extensions; mailing list; contact authors; tutorial; examples; wiki	http://sitemaker.umich.edu/soar/home
Sugarscape	Java 2 SDK or (Internet Explorer 5.x or greater AND the Java 2 Runtime Environment	API	http://sugarscape.sourceforge.net/

	(JRE))		
Swarm	Windows; Linux; Macintosh OS X	Wiki; tutorials; examples; documentation; FAQ; selected publications; mailing lists	http://www.swarm.org/
VSEit	To run simulations: a Java enabled internet browser like Netscape Navigator or Microsoft Explorer. VSEit is known to run under Netscape Navigator 4.06 or higher, on Windows 95/98 and Windows NT; to develop simulations: any Java platform supporting Java 1.1.7.	Examples; users guide; defect list	http://www.vseit.de

ZEUS	Windows 95;
	Windows 98;
	Windows NT;
	Windows 2000;
	Windows XP;
	Linux; BSD;
	UNIX-like OSes;
	Solaris
		Documentation; author contact	http://labs.bt.com/projects/agents/zeus/

[3] An important note for the reader is that we tried to look at each toolkits as completely and comprehensively as possible. However, our study is not complete. In places where it is not complete, it is because the authors have not specified the complete granularity of the platforms with respect to different operating systems.

[4] Tryllian can offer paid support for running the ADK on IBM mainframes or AS/400 machines. Apple's OS X is not supported, but part of the development of the ADK is done on OS X. 5 If you need to run the ADK on JDK 5.0, please contact Tryllian. The ADK will not work out of the box with version 5.0 of the JDK. The ADK has also been tested with Bea's JRockit JVM and with IBM's JVM; both appear to support running the ADK.

References

AGENT BUILDING AND LEARNING ENVIRONMENT (ABLE) Website. http://www.alphaworks.ibm.com/tech/able.

AGENTBUILDER LITE/PRO Website. . http://www.agentbuilder.com/Documentation/Lite/.

AGENTSHEETS Website. http://www.agentsheets.com/research/c5/index.html.

ANYLOGIC Website. http://www.xjtek.com/anylogic/.

ASCAPE Website. http://ascape.sourceforge.net/index.html#Introduction.

BALAN, G, Cioffi-Revilla, C, Luke, S, Panait, L and Paus, S (2003) Mason: A Java Multi-Agent Simulation Library. *Proceedings of the Agent 2003 Conference*, October 2003. pp59-64.

BELLIFEMINE, F, Caire, G, Poggi, A and Rimassa, G (2003) JADE - A White Paper. *TILAB EXP in search of innovation* 3 (3) http://jade.tilab.com/papers-exp.htm.

BELLIFEMINE, F, Poggi, A and Rimassa, G (1999) JADE - A FIPA-Compliant Agent Framework *Proceedings of the Practical Applications of Intelligent Agents,* 1999. pp97-108.

BIGUS, J, Schlosnagle, D, Pilgrim, J, Mills III, W and Diao, Y (2002) ABLE: A Toolkit for Building Multiagent Autonomic Systems. *IBM Systems Journal*, September 2002. pp350-371.

BONABEAU E (2002) Agent-based Modeling: Methods and Techniques for Simulating Human Systems. *Proceedings of the National Academy of Sciences of the United States of America*, May 2002. pp7280-7287.

BORDINI R, Carlos da Rocha Costa, A, Hubner, J, Moreira, A, Okuyama, F and Vieira, R (2005) MAS-SOC: a Social Simulation Platform Based on Agent-Oriented Programming. *Journal of Artificial Societies and Social Simulation* 8 (3) 7 http://jasss.soc.surrey.ac.uk/8/3/7.html.

BORSHCHEV A, Karpov Y, and Kharitonov, V (2002) Distributed Simulation of Hybrid Systems with AnyLogic and HLA. *Future Generation Computer Systems,* May 2002. pp 829-839.

BORSHCHEV A and Filippov A (2004) From System Dynamics and Discrete Event to Practical

Agent Based Modeling: Reasons, Techniques, and Tools. *22nd International Conference of the System Dynamics Society,* July 2004. pp959-966.

BRAIN AND HEAD MODELLING SYSTEM (BRAHMS) Website. http://sourceforge.net/forum/forum.php?forum_id=685983.

BRASSEL K (2001) Flexible Modelling with VSEit, the Versatile Simulation Environment for the Internet. *Journal of Artificial Societies and Social Simulation* 4 (3) 10 http://www.soc.surrey.ac.uk/JASSS/4/3/10.html.

BREVE Website. http://www.spiderland.org/node/2602.

CARVALHO J (2000) Using AgentSheets to Teach Simulation to Undergraduate Students. *Journal of Artificial Societies and Social Simulation* 3 (3) 2 http://www.soc.surrey.ac.uk/JASSS/3/3/forum/2.html.

CASTLE C and Crooks A (2006), Principles and Concepts of Agent-Based Modelling for Developing Geospatial Simulations. *Working paper 110, Centre for Advanced Spatial Analysis, University College London*, September 2006.

COLLIER N and North M (2004) Repast for Python Scripting. *Proceedings of the Agent 2004 Conference*, October 2004. pp231-237.

COMMON-POOL RESOURCES AND MULTI-AGENT SYSTEMS (CORMAS) Website. http://cormas.cirad.fr/indexeng.htm.

CONTE, R, Gilbert, N and Simao Sichman, J (1998) MAS and Social Simulation: A Suitable Commitment. *Lecture Notes in Computer Science 1534*. pp. 1-9.

COUGAAR Website. http://www.cougaar.org/.

DEX Website. http://dextk.org/dex/index.html.

DISTRIBUTED OPERATOR MODEL ARCHITECTURE (D-OMAR) Website. http://omar.bbn.com/.

ECHO Website. http://www.santafe.edu/~pth/echo/.

ECJ Website. http://cs.gmu.edu/~eclab/projects/ecj/.

FORREST S and Jones T (1994) Modeling Complex Adaptive Systems with Echo. *Complex Systems: Mechanism of Adaptation,* 1994. pp 3-21.

FRAMEWORK FOR AGENT-BASED MODELLING WITH JAVA (FAMOJA) Website. http://www.usf.uos.de/projects/famoja/.

FREEBSD.ORG Website. http://www.freebsd.org/copyright/license.html.

FREIWALD U and Weimar J (2000) JCA-Sim - A Java System For Simulating Cellular Automata. *Proceedings of the Fourth International Conference on Cellular Automata for Research and Industry: Theoretical and Practical Issues on Cellular Automata,* 2000. pp47-54.

GILBERT N and Bankes S (2002) Platforms and Methods for Agent-based Modeling. *Proceedings of the National Academy of Sciences.* May 2002. pp197-198.

GNU Homepage. http://www.gnu.org/licenses/licenses.html.

GULVAS, L, Kosik, T and Corliss, J (1999) The Multi-Agent Modelling Language and the Model Design Interface. *Journal of Artificial Societies and Social Simulation* 2 (3) 8 http://www.soc.surrey.ac.uk/JASSS/2/3/8.html.

HELSINGER, A, Thome, M and Wright, T (2004) Cougaar: A Scalable, Distributed Multi-Agent Architecture. *IEEE INTERNATIONAL CONFERENCE ON SYSTEMS MAN AND CYBERNETICS* 2, October 2004. pp1910-1917.

HERMANS B (1997) Intelligent Software Agents on the Internet: An Inventory of Currently Offered Functionality in the Information Society and a Prediction of (Near) Future Developments. *First Monday* 2 (3) 123 http://www.firstmonday.dk/issues/issue2_3/ch_123.

IGEN Website. http://www.cognitiveagent.com/.

JADE Website. http://jade.tilab.com.

JADE'S SIM++. no longer available online.

JAS Website. http://jaslibrary.sourceforge.net/.

JAVA AUCTION SIMULATOR API (JASA) Website. http://sourceforge.net/projects/jasa/.

JAVA ENTERPRISE SIMULATOR (JES) Website. http://web.econ.unito.it/terna/jes/.

JCA-SIM Website. http://www.jweimar.de/jcasim/jcasim.html.

JECHO Website. http://www.brianmcindoe.com/.

JESS Website. http://herzberg.ca.sandia.gov/jess/.

KOEHLER M and Tivnan B (2005) Clustered Computing with NetLogo and Repast J: Beyond

Chewing Gum and Duct Tape. *Proceedings of the Agent 2005 Conference*, October 2005. pp43-54.

LABORATORY FOR SIMULATION DEVELOPMENT (LSD) Website. http://www.business.aau.dk/lsd/lsd.html.

LESZCZYNA R (2004) Evaluation of Agent Platforms. *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems,* 2004. pp1063-1070.

LOMOW G and Baezner D (1991) A Tutorial Introduction to Object-Oriented Simulation and Sim++. *Proceeedings of the 1991 Winter Simulation Conference*, December 1991. pp157-163.

LUKE, S, Cioffi-Revilla, C, Panait, L and Sullivan, K (2004) MASON: A New Multi-Agent Simulation Toolkit. *Swarmfest 2004, Eighth Annual Swarm Users/Researchers Conference,* May 2004.

LUSTICK I (2002) PS-I: A User-Friendly Agent-Based Modeling Platform for Testing Theories of Political Identity and Political Stability. *Journal of Artificial Societies and Social Simulation* 5 (3) 7 http://jasss.soc.surrey.ac.uk/5/3/7.html.

MACAL C (2004) Agent-Based Modeling and Social Simulation with Mathematica and Matlab *Proceedings of the Agent 2004 Conference,* October 2004. pp185-204.

MACAL C and Howe T (2005) Linking Repast and Computational Mathematics Systems: Mathematica and Matlab. *Proceedings of the Agent 2005 Conference*, October 2005. pp5-24.

MACSTARLOGO. Available from the StarLogo webpage: http://education.mit.edu/starlogo/.

MULTI-AGENT SYSTEM (MAGSY) Website. http://www-ags.dfki.uni-sb.de/~kuf/magsy.html.

MASON Website. http://cs.gmu.edu/~eclab/projects/mason/.

MATRIX LABORATORY (MATLAB) Website. http://www.mathworks.com/access/helpdesk/help/techdoc/index.html?/access/helpdesk/help/techdoc/learn_matlab/f48955.html&http://www.mathworks.com/access/helpdesk/help/tec-doc/matlab_product_page.html.

MCINDOE B (2005) jECHO: A New Implementation Of Holland's Complex Adaptive Systems Model With The Aim Of Seeking Answers To Some Fundamental Open Questions. *Master of Science Dissertation,* February 2005. http://www.brianmcindoe.com.

MICRO-UND MULTILEVEL MODELLING SOFTWARE (MIMOSE) Website. http://www.uni-koblenz.de/~moeh/projekte/mimose.html.

MODULECO Website. http://www.cs.manchester.ac.uk/ai/public/moduleco/.

MOHRING M (1995) Social Science Multilevel Simulation with MIMOSE. *Social Science Microsimulation*, May 1995. pp123-137.

MULTI AGENT DEVELOPMENT KIT (MADKIT) Website. http://www.madkit.org/.

MULTI-AGENT MODELING LANGUAGE (MAML) Website. http://www.maml.hu/.

MULTI-AGENT SIMULATIONS FOR THE SOCIAL SCIENCES (MAS-SOC) Website. http://inf.ufrgs.br/massoc.

MULTIMODELING OBJECT-ORIENTED SIMULATION ENVIRONMENT (MOOSE) Website. http://www.cise.ufl.edu/~fishwick/moose.html.

NAJLIS R and North M (2004) Repast for GIS. *Proceedings of the Agent 2004 Conference*, October 2004. pp255-259.

NETLOGO Website. http://ccl.northwestern.edu/netlogo/models/.

NGUYEN, G, Dang ,T, Hluchy, L, Laclavik, M, Balogh, Z and Budinska, I (2002) Agent Platform Evaluation and Comparison. II-SAS, Pellucid EU 5FP IST-2001-34519 RTD, Technical report, June 2002.

NORTH, M, Howe, T, Collier, N and Vos, J (2005) The Repast Simphony Development Environment. *Proceedings of the Agent 2005 Conference*, October 2005. pp159-166.

NORTH, M, Howe, T, Collier, N and Vos, J (2005) The Repast Simphony Runtime System. *Proceedings of the Agent 2005 Conference*, October 2005. pp151-158.

NORTH, M, Sydelko, P, Vos, J, Howe, T and Collier, N (2006) Legacy Model Integration with Repast Simphony. *Proceedings of the Agent 2006 Conference*, October 2006. pp95-106.

OBJECT BASED ENVIRONMENT FOR URBAN SIMULATION (OBEUS) Website. http://eslab.tau.ac.il/Research/OBEUS/default.aspx.

OPENSTARLOGO Website. http://education.mit.edu/openstarlogo/.

ORIS Website. http://www.enib.fr/~harrouet/.

PARKER M (2001) What is Ascape and Why Should You Care? *Journal of Artificial Societies and*

Social Simulation 4 (1) 5 http://jasss.soc.surrey.ac.uk/4/1/5.html.

PHAN D and Beugnard A (2001) Moduleco, a Multi-Agent Modular Framework for the Simulation of Network Effects and Population Dynamics in Social Sciences, Markets & Organisations. *8th Rencontre Internationale ACSEG* (Approches Connexionnistes en Sciences Economiques et de Gestion), October 2001.

POLITICAL SCIENCE-IDENTITY (PS-I) Website. http://ps-i.sourceforge.net/.

QUICKSILVER (NOW CALLED OMONIA) Website. http://www.xlog.ch/omonia/.

RAILSBACK, S, Lytinen, S and Jackson, S (2006) Agent-Based Simulation Platforms: Review And Development Recommendations. *Simulation*, September 2006. pp609-623.

RECURSIVE POROUS AGENT SIMULATION TOOLKIT (REPAST) Website. http://repast.sourceforge.net.

REPAST.NET Website. http://repast.sourceforge.net/repast_3/download.html.

REPASTJ Website. http://repast.sourceforge.net/repast_3/download.html.

REPASTPY Website. http://repast.sourceforge.net/repast_3/download.html.

REPASTS Website. http://repast.sourceforge.net.

REPENNING, A, Ioannidou, A and Zola, J (2000) AgentSheets: End-User Programmable Simulations *Journal of Artificial Societies and Social Simulation* 3 (3) forum 1 http://www.soc.surrey.ac.uk/JASSS/3/3/forum/1.html.

SAMUELSON D and Macal C (2006) Agent-Based Simulation Comes of Age. *OR/MS Today*, August 2006. pp34-38.

SERENKO A and Detlor B (2002) Agent Toolkits: A General Overview of The Market and an Assessment of Instructor Satisfaction with Utilizing Toolkits in the Classroom. *Working Paper #455*.

SHELL FOR SIMULATED AGENT SYSTEMS (SESAM) Website. http://www.simsesam.de/.

SIERHUI, M, Clancey, W and van Hoof, R (2007) BRAHMS A Multi-agent Modelling Environment for Simulating Work Processes and Practices. *International Journal of Simulation and Process Modelling*, 2007. pp134-152.

SIMAGENT Website. http://www.cs.bham.ac.uk/research/projects/poplog/packages/simagent.html.

SIMBIOSYS Website. http://www.lucifer.com/~david/SimBioSys/.

SIMPACK Website. http://www.cis.ufl.edu/~fishwick/simpack/simpack.html.

SIMPLUSPLUS Website. http://www.simplusplus.com/.

SOAR Website. http://sitemaker.umich.edu/soar/home.

SONNESSA M (2003) JAS: Java Agent-based Simulation Library. An Open Framework for Algorithm-Intensive Simulation. *Industry And Labor Dynamics: The Agent-Based Computational Economics Approach*, October 2003. pp43-56.

SPATIAL MODELING ENVIRONMENT (SME) Website. http://www.uvm.edu/giee/SME3/.

STANDISH R and Leow R (2003) EcoLab: Agent Based Modeling for C++ Programmers. *Proceedings SwarmFest,* April 2003.

STARLOGO TNG (THE NEXT GENERATION) Website. http://education.mit.edu/starlogo-tng/index.htm.

STARLOGO Website. http://education.mit.edu/starlogo/.

STARLOGOT Website. http://ccl.northwestern.edu/cm/starlogoT/.

STRICTLY DECLARATIVE MODELING LANGUAGE (SDML) Website. http://cfpm.org/sdml/.

SUGARSCAPE Website. http://sugarscape.sourceforge.net/.

SWARM Website. http://www.swarm.org.

TATARA, E, North, M, Howe, T, Collier, N and Vos, J (2006) An Introduction to Repast Simphony Modeling Using a Simple Predator-Prey Example. *Proceedings of the Agent 2006 Conference on Social Agents*, October 2006. pp1-11.

TERNA P (2001) Creating Artificial Worlds: A Note on *Sugarscape* and Two Comments. *Journal of Artificial Societies and Social Simulation* 4 (2) 9 http://www.soc.surrey.ac.uk/JASSS/4/2/9.html.

THORNGATE W (2000) Teaching Social Simulation with Matlab *Journal of Artificial Societies and Social Simulation* 3 (1) forum 1 http://jasss.soc.surrey.ac.uk/3/1/forum/1.html.

TISSEAU J (2001) Virtual Reality - In Virtuo Autonomy Computer Science Thesis of the

University of Rennes, December 2001. pp61-74.

TISUE S and Wilensky U (2004) NetLogo: Design and Implementation of a Multi-Agent Modeling Environment. *Proceedings of the Agent 2004 Conference*, October 2004. pp161-184.

TOBIAS R and Hofmann C (2004) Evaluation of free Java-libraries for social-scientific agent based simulation. *Journal of Artificial Societies and Social Simulation* 7 (1) 6 http://jasss.soc.surrey.ac.uk/7/1/6.html.

TRYLLIAN AGENT DEVELOPMENT KIT Website. http://www.tryllian.com.

VSEIT Website. http://www.vseit.de.

VOS J and North M (2004) Repast.net. *Proceedings of the Agent 2004 Conference*, October 2004. pp239-254.

XU H and Shatz S (2003) ADK: An Agent Development Kit Based on a Formal Design Model for Multi-Agent Systems. *Automated Software Engineering*, October 2003. pp337-365.

ZAUNER, G, Leitner, D and Breitenecker, F (2007) Modeling Structural - Dynamics Systems in MODELICA/Dymola, MODELICA/Mosilab and AnyLogic. *Proceedings of the 1st International Workshop on Equation-Based Object-Oriented Languages and Tools*, July 2007. pp 99-110.

ZEUS Website. http://labs.bt.com/projects/agents/zeus/ .

Return to Contents of this issue