

# Searchable Taxonomies of Agent Based Modeling Toolkits

Cynthia Nikolai, Gregory Madey  
University of Notre Dame  
Computer Science & Engineering  
College of Engineering  
Notre Dame, IN, 46556 USA  
[cnikolai@nd.edu](mailto:cnikolai@nd.edu), [gmadey@nd.edu](mailto:gmadey@nd.edu)

**Keywords:** agent based modeling, toolkits, toolkit search engine

## Abstract

In this paper, we present a web-based search appliance for Agent Based Modeling (ABM) toolkit selection. This search appliance enables users to query a comprehensive ABM repository to determine toolkits that are appropriate for their project. It searches the repository according to the criteria set forth by the user, and then it displays the results in a series of user friendly and interactive graphical and textual tables. It also serves as a potential client base for further research into ABM toolkits; Once on the site, the users are asked if they would like to participate in a toolkit survey, thereby furthering ABM research and also feeding back to improve the web tool itself.

## 1. INTRODUCTION

Agent Based Modeling is a simulation paradigm in which one uses bottom up techniques to model large numbers of very simple agents. The behavior of agents is specified via a set of individual rules and then the agents are “let loose” in a simulation environment. As the agents interact with each other, researchers examine the emergent behavior of the system and try to draw inferences and make conclusions [Borshchev 2004].

In this research, we address ABM toolkits. There are several excellent ABM toolkit surveys available [Castle and Crooks 2006][Railsback et al. 2006][Tobias and Hofmann 2004]. However, there also are several limitations. First, these surveys are limited to four or five mainstay and historically similar toolkits. Moreover, these surveys are presented from the point of view of one or two communities of interest [Castle and Crooks 2006][Railsback et al. 2006][Tobias and Hofmann 2004]. A second limitation is that since this type of modeling and simulation spans multiple disciplinary fields, the surveys that are available are scattered across multiple domains (e.g. computer science, social science, artificial intelligence [Castle and Crooks 2006][Railsback et al. 2006][Tobias and Hofmann 2004]. Researchers have to scour many different sources to come up with a global picture. Finally, all surveys are text

based, and there is no centralized place online where people can make use of that knowledge.

In this research, we have addressed these limitations through a comprehensive ABM toolkit survey, in which we have examined many of the less well known platforms in addition to the most popular ones [Nikolai and Madey 2009]. Next we assimilated this information into a relational database. Now we take the next step and create a search engine that quickly and efficiently helps users from any discipline to find a list of toolkits that are applicable for their research.

This paper is structured as follows: First, we discuss our general methodology. Next, we discuss several agent based modeling taxonomies based on our research. After that, we outline the content of the search engine, and we describe the basic system workflow. Then, we examine each of the search engine subsystems and the underlying databases. Next, we show several example searches using the search engine. Following this, we discuss the limitations of this work. Finally, we finish with a few concluding remarks and references.

## 2. METHODOLOGY

The research underlying this repository is a culmination of on-going research into ABM toolkits over the last 12 months. We began by gathering a comprehensive list of agent based toolkits available and that are being used in some fashion for ABM purposes (see table 1).

**Table 1.** List of Toolkits

ABLE	MAS-SOC
AgentBuilder Lite/Pro	Matlab
ADK	MIMOSE
AgentSheets	Moduleco
AnyLogic	StarLogo and family
Ascape	NetLogo
Brahms	OBEUS
Breve	oRIS

Cormas	PS-I
Cougaar	FAMOJA
DeX	Quicksilver (now called Omonia)
DOMAR	Repast and family
ECHO	SDML
jEcho	Jade's Sim++
ECJ	SimPlusPlus
iGen	SimAgent (aka sim_agent)
JADE	SimBioSys
JAS	Moose
JASA	SimPack
JCA-Sim	SME
jES	SeSAm
JESS	SOAR
LSD	Sugarscape
Madkit	Swarm
MAGSY	VSEit
MAML	Zeus
Mason	

These include any platforms that are available, 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 available information 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. Then, we sorted through all the information and created various taxonomies based on major classifications. Next, we created a database repository of these classifications. Finally, we implemented an online search engine to allow users to access the information in a variety of ways in a centralized location. That is, users can input a range of desired toolkit characteristics, and the tool returns a list of candidate platforms that contain those characteristics. There are four main characteristics available across the spectrum of toolkits from which the users can choose: programming language required to create a model or simulation, operating system required to run the toolkit, type of license governing the platform, and primary domain for which the toolkit is intended. We chose these characteristics because they are usually the first features that one looks at when choosing a toolkit for a project [Castle and Crooks 2006][Leszczyna 2004] [Tobias and Hofmann 2004].

### 3. TAXONOMIES

In this section, we give the reader a brief introduction to the various agent based modeling taxonomies and several conclusions based on these taxonomies.

#### 3.1. Programming Language

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. 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.

By far, the main programming language most models have adopted is Java. About 42% of the platforms employ Java as their primary programming language. See table 2.

**Table 2.** List of Toolkits That Support Java Programming

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

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. See [Nikolai and Madey 2009] for the complete set of taxonomies.

### 3.2. Type of License

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 the licensee to release the source code of commercial extensions to the platform (freebsd.org website).

We have organized the licenses into four main branches. 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%). See table 3 for a list of toolkits that are classified as free under open source licenses. 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 already is incorporated into the toolkit or for additional features that may be incorporated into the toolkit by the user. See [Nikolai and Madey 2009] for the complete set of taxonomies.

**Table 3.** List of Toolkits That Are Classified As Free Under Open Source Licenses

Type of License	Toolkit
Open Source (uncategorized <sup>1</sup> )	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

<sup>1</sup>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.

### 3.3. Operating System

The third category we examine is the operating system on which the toolkits run. The majority of toolkits run on Windows (see table 4) 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. See [Nikolai and Madey 2009] for the complete set of taxonomies.

**Table 4.** Toolkits that Run on Various Windows Operating Systems<sup>2</sup>

Operating System	Toolkit
Windows (version not specified <sup>2</sup> )	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

<sup>2</sup>An important note for the reader is that we tried to look at each toolkit as 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 also may 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.

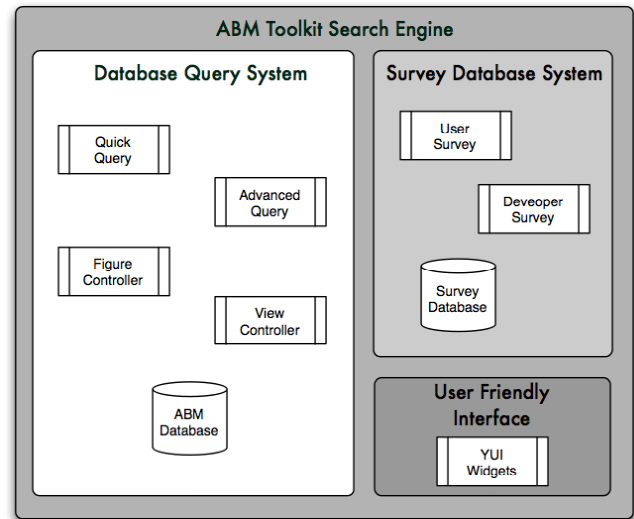
### 3.4. Primary Domain

In this section, we examine various domains for which the toolkits are specialized. Many of the toolkits specifically are tailored for particular domains, and many are general purpose toolkits that can be used for a variety of domains. The main domains are depicted in table 5. 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 than just their primary. The secondary domains, however, 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 it will bring to the fore potential toolkits that the user otherwise may not have considered. See [Nikolai and Madey 2009] for the complete set of taxonomies.

**Table 5.** List of Documented Primary Domains Across the ABM Spectrum

Applied simulations/electronic CAD	Enterprises ➤ Single enterprise ➤ Systems of enterprises
Artificial Intelligence (general)	Evolutionary computing

purpose)		➤ Teaching computer simulation	➤ Social systems
➤ Machine learning and reasoning	General purpose agent based	➤ Using object oriented principles	Testing Base24 applications
➤ Social sciences	➤ Multi-agent systems with agent based simulation layer	➤ Teaching programming techniques to students new to simulation	Urban simulation
➤ 3D simulations ➤ multi-agent systems	➤ General-purpose parallel applications	➤ Scientific and engineering math and computation; data analysis, exploration, and visualization	
➤ Human-like intelligent agents	Human performance modeling		
Biology	➤ Training systems	4. <b>SEARCH ENGINE OVERVIEW</b>	The web tool consists of 3 main components: (1) a database query system (2) a survey database system (3) a user-friendly user-interface (UFI). (see figure 1).
Cellular automata	➤ Performance support systems		
Computational economics/Auction mechanisms	Multi-agent systems (general purpose)		
Distributed simulations	➤ Large scale distributed		
➤ Large scale	➤ Complex environments		
➤ Mobile (distributed) agents	➤ Distributed		
➤ Small scale embedded	➤ Decision-making in complex environments (with limited rationality)		
➤ Virtual reality	➤ Organizational processes		
➤ Highly distributed, scalable, reliable, survivable applications	Natural resources management		
Ecological modeling	Natural Sciences		
Education	Political phenomena		
➤ Artificial Intelligence	Rule engine and scripting environment		
➤ For students to model the behavior of decentralized systems	Simulating organizational processes		
➤ Implementing software agents	Social sciences		
➤ Learning (including explanation based learning)	➤ Help beginning users get started authoring models		



**Figure 1.** Search Engine System

The database query system consists of a form that asks the users for a query regarding ABM toolkits. It contains both a quick query and an advanced query capability. The quick query contains four fields. Each field consists of a pull-down menu for easy selection of a one-item database query. The database query system also contains a tab controller which manages user tabs, onto which results of queries are appended to the document object model (DOM). Finally, the database query system contains a figure controller and a view controller that allow for user interaction with the query results. Underlying the system is a PostgreSQL database.

The Survey Database System consists of a user survey,

a developer survey and an underlying PostgreSQL database. The ABM questionnaires were developed from current issues outlined in the literature, and the questions themselves were based on survey design recommendations from the literature. We structured the questions such that we addressed the limitations of each type of question to reduce its inherent biases. We used recommended techniques such as writing clear questions, using syntax and linguistics to facilitate understanding and respondent recall, and developing unscaled response lists to minimize question bias, memory error, and measurement error. We used odd numbered Likert scales to allow for neutrality in decisions by the respondent. We quantified all scaled values, anchored our scaled lists, and incorporated escape choices for the respondent, including areas for additional comments. Finally, in order to obtain a higher response rate, we have incorporated a pledge of anonymity.

The user friendly interface consists of a series of Yahoo User Interface (YUI) interactive widgets and meta-information.

#### 4.1. Under the Hood

The main workflow is depicted in figure 2. First, the user opens a browser and requests the main page (abmsearchengine.php) from the web server. The server then does a query to the ABM database to retrieve pull-down menu items for the quick query fields. Next, it contacts the yahoo server (http://yui.yahooapis.com) and registers tabView widgets for the user interface. After that, the server accesses a file to create a unique ID for the session. This ID is sent back to the user and is stored in the underlying JavaScript. In addition to rendering a page count, the user ID later will be used as the user's unique ID for the survey database system. Finally, the server uses JavaScript, XHTML, PHP, and YUI to render the main page. From there, the user can choose to perform a quick query, an advanced query, or take an ABM survey.

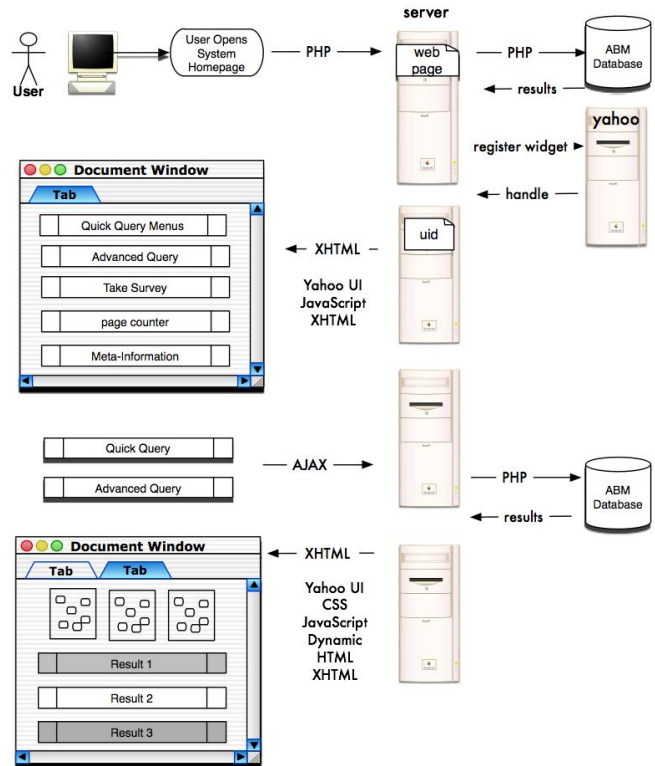


Figure 2. Search Engine Workflow

#### 4.2. Database Query System

The database query system consists of a form that allows users to query the ABM toolkits based either on predefined criteria or criteria set forth by the user (see figure 3). It

QuickTime™ and a  
TIFF (uncompressed) decompressor  
are needed to see this picture.

Figure 3. Main Page of the Search Engine

contains both a quick query and an advanced query capability. The quick query has four fields. The main categories available to the user are: language required for programming, toolkit domain, type of license, and operating systems supported. Each field in the quick query contains a pull-down menu for easy selection of a one-item database query. The user initiates the query by clicking on the “Go!” button associated with the quick query form. After selecting ‘Go!,” the query returns a join of the fields as constraints in the where clause of an SQL select statement. If the user does not select any elements in the four quick fields, the query returned is “select \* from ABM table” and “select \* from image\_table where tag = all\_languages or tag = all\_domains or tag = all\_platforms or tag = all\_licenses.” The system makes an asynchronous JavaScript call to query the database and return the results. Internally, when the results are ready, the system creates a new tab and appends it to the current DOM. The system next activates the new tab and then appends the query results to the new tab’s DOM. Each new query results in a new tab.

This is a way of maintaining session state so the user can compare different queries. We decided to maintain this state on the client because the database will be used by a large number of clients, and this is an efficient way to maintain session state for a large number of users. The results are returned in the form of a table of figures (see figure 4) followed by a table of textual results sorted alphabetically (see figure 5). The figures are displayed as a set of thumbnails in a table at the top of the page. Since some of the figures are dense and may be difficult to see, we included an option for the user to view a larger version of the image. When the user double-clicks on the image, the figure controller, using Dynamic HTML and JavaScript, enlarges the image. When the user clicks on the image one more time, it returns the image to its normal size.

The user also can customize and interact with the textual query results via a view controller. The default view returned from the query is the summary view, which is a condensed view of the toolkit. If the user would like to see more detailed information, the user can expand a detailed view by clicking on the appropriate button. Using dynamic HTML, CSS, and by modifying the DOM, we add additional fields to the summary view. If the user no longer wants to see the detailed information, the user can collapse the view. Internally, we accomplish this by performing an additional AJAX query to the database and then appending the information to the DOM. To collapse the detailed view, we again traverse the DOM and delete the detailed information. Finally, if the user wants to visit the main toolkit webpage to get first-hand information, the user can click on the webpage link, which is returned in an XHTML HREF tag. When the user clicks on the link, it opens the

toolkit webpage in another window. When the user is finished reviewing the results, the user can go back to the home page and initiate more queries. Each new query is appended to the DOM as a new tab.

QuickTime™ and a  
TIFF (Uncompressed) decompressor  
are needed to see this picture.

**Figure 4.** Graphical Query Results

QuickTime™ and a  
TIFF (Uncompressed) decompressor  
are needed to see this picture.

**Figure 5.** Textual Query Results

The advanced query also is a form, and it has the same workflow as the quick query. However, it allows users to enter advanced queries. Advanced queries can be any valid SQL query to the database. For example, if the user wants to search for toolkits that use either Java or C, they would have to use an advanced query as specified below.

```
select * from abm_table where language ~* 'C' or
language ~* 'Java'
```

### 4.3. Survey Database System

This system consists of a form that contains a toolkit survey. If the user elects to participate in the ABM survey, he/she can click on the button “take ABM survey.” This renders a page containing survey instructions. After reading the instructions, the user is asked which category of toolkit expert he/she is: a user or a developer. Depending on the response, the user is presented either with a user survey or a developer survey. We use JavaScript to process the response on the client in order to save unnecessary journeys to and from the server.

The questionnaire is broken into three pages with submission between each page. When the user submits the form, the system stores the answers in a database for later analysis. The survey workflow is depicted in figure 6. The reader may have noticed the user id (UID) that is passed between documents throughout the workflow. This is a hidden form field, and it contains the unique number that was created when the user first opened the main page. In addition to rendering the page counter, this number becomes the user’s ID for the survey. We carry this state with us until we finish with the survey and enter the data into the database.

### 4.4. User Friendly Interface (UFI)

This system consists of a user-friendly interface to the database query system and the survey database system. The UFI mainly consists of XHTML, AJAX, JavaScript and YUI widgets. The YUI was used to create multiple tabs for the user.

### 4.5. Databases

There are two main database systems underlying the search engine (see table 2). For the database query system, we have a series of tables for each category of interest in the quick query. We also have a main ABM table to store additional toolkit characteristics. Finally, we have an image table that stores the filename and corresponding search tags of our ABM figures. For the survey database system, we have a table for each type of question. There are four main question types; questions which require the user to input text in a text box (text questions), questions that require the user to choose from among a series of mutually exclusive choices (radio questions), questions that require the user to select all valid choices that are present (checkbox questions), and finally, questions for which the answer is open ended or for which the user may enter additional comments concerning the questions (comments questions). Each question is indexed in a corresponding table according to its type. Because we have different surveys for users and for developers, we have a complementary set of tables to store the user results and the developer results. Upon completing the survey, the user responses are indexed by the

unique user ID and then stored in the appropriate database tables.

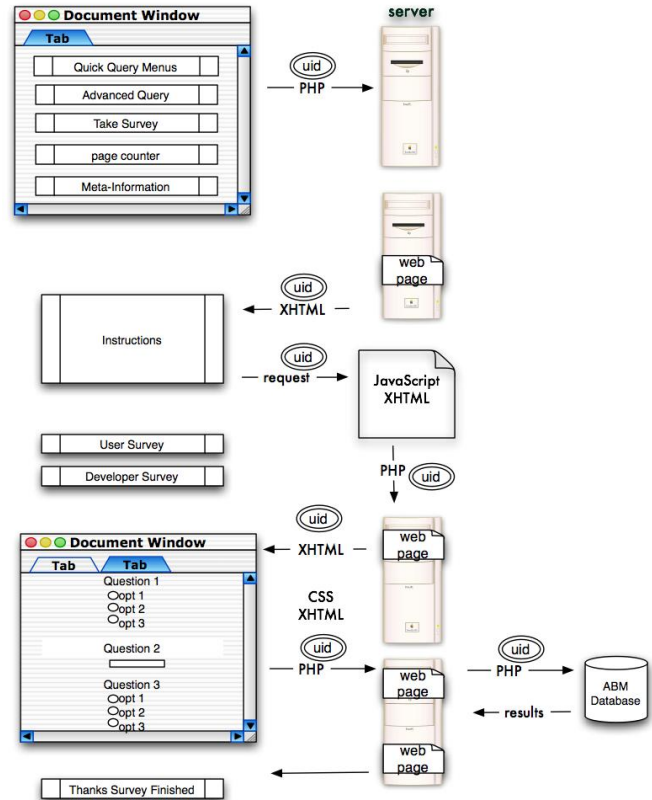


Figure 6. Survey Database System

Table 2. System Database Tables

Database Query System	Survey Database System	
Domain_table ID, domain	User Tables Text ID,q1,q2,q3,...qn	Developer Tables dText ID,q1,q2,q3,...qn
Language_table ID, language	Checkbox ID,q1,q2,q3,...qm	dCheckbox ID,q1,q2,q3,...qm
License_table ID, license	Radio ID,q1,q2,q3,...qp	dRadio ID,q1,q2,q3,...qp
Platform_table ID, platform	Comments ID,q1,q2,q3...qt	dComments ID,q1,q2,q3...qt
ABM_table ID, name, platform, domain, language, license, website, notes		
Image_table ID, filename, tag		



## 5. DEPLOYMENT

We have successfully deployed our system by posting it on our web server, and it can be accessed at <http://agent.cse.nd.edu/abmsearchengine.php>.

## 6. RESULTS: EXAMPLE TOOLKIT SEARCH

In this section, we outline an example toolkit search for a quick query and an advanced query.

### 6.1. Quick Query

Suppose we have a requirement to program a model and simulation in Java, and a general-purpose agent based toolkit. Due to budget constraints, we need the toolkit to be free, but we also need it to be open source. We input the following parameters into the search engine. domain = "General Purpose Agent Based" license = "Free: Open Source" and Programming Language = "Java." Because we have no requirement concerning the platform on which our toolkit runs, we leave Platforms Supported = "\*". After hitting "Go," we have the following results.

JAS, Madkit, RepastS, and Swarm

### 6.2. Advanced Query

Suppose we have a requirement to program a model or simulation in Java or C, and due to management constraints, we need a GPL-governed toolkit. This requires an advanced query. We input the following simple SQL statement into the advanced query.

```
select * from abm_table where (language ~* 'C'  
or language ~* 'Java') and license ~* 'GPL'
```

We obtain the following results.

Tryllian Agent Development Kit (ADK),  
FAMOJA, JADE, JAS, Madkit, Quicksilver  
(now called Omonia)

## 7. LIMITATIONS

### 7.1. Technological

The first limitation of this system is that the user has to enable JavaScript. This requires some element of trust and risk from the user, and the user may not be willing to accept that risk. In Internet Explorer, the user also has to enable ActiveX controls, and for Netscape, the user has to put the website in his/her trusted sites.

Another technological limitation is a lack of good feedback to the user when an incorrect query is entered. Currently, we provide for a standard error returned by the SQL database. This can be difficult to understand, however, if one is not familiar with SQL. We would like to update this to reflect better feedback.

### 7.2. Research

There also are several limitations stemming from the ABM research itself. First, it is difficult to draw the line between ABM toolkits and multi-agent system (MAS) toolkits. Although we are focused in this research on ABM, the distinction is not clear because some toolkits with ABM contain underlying artificial intelligence and the ability to create more advanced agents, and some MAS are able to create larger numbers of simple agents.

Another limitation is in the questionnaire; it is still in development. Also, since this is a voluntary survey, there is no way to guard against incomplete or biased information. We do not check that all questions are answered because we do not want to hassle the user. We also would like to guard against biases which occur because the user already has taken the survey. We store this information in a cookie, but if the user disables cookies, then there is no way to counter it.

A third limitation is the maintainability of the system. While we do have the most current database repository underlying the search engine, the information can easily grow stale. We will make some attempt to keep the underlying source of information current in the near future. However, it should be noted that this tool may prove most useful now and in the next few years.

Finally, because we are waiting on preliminary survey responses, the detailed information is only a template at this time. Once we obtain survey results, we will incorporate them into our detailed summary view.

## 8. ON-GOING WORK

As a complement to this tool, we also have created a corresponding wiki in wikipedia based on this research. Besides 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 located at the following URL: [http://en.wikipedia.org/wiki/ABM\\_Software\\_Comparison](http://en.wikipedia.org/wiki/ABM_Software_Comparison). We invite the ABM community to participate in expanding this research further. In the future, we would like to incorporate the results of the wiki into our repository as well.

## 9. CONCLUSION

In this work, we have outlined our comprehensive online ABM toolkit search engine. We began by describing our methodology and the underlying source of our ABM data repository. Next we presented several agent based modeling taxonomies based on our research. After that, we described the search engine and each of its subsystems.

Then we gave a sort example of the search engine quick query and advanced query. Finally, we concluded with a few limitations and directions for future work.

## References

Agent Based Modeling Search Engine. Available at <http://agent.cse.nd.edu/abmsearchengine.php>.

Arsham, H., 2002, "Questionnaire Design and Survey Sampling." Available at <http://www.sysurvey.com/tips/arsham.htm>.

Borshchev, 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.

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

Creative Research Systems. 2006. "Survey Design." available at <http://www.surveysystem.com/sdesign.htm>.

Elmasi, R., and S. Navathe. 2007. "Fundamentals of Database Systems." 5th edition, Pearson Education, Inc.

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). [http://www.firstmonday.dk/issues/issue2\\_3/ch\\_123](http://www.firstmonday.dk/issues/issue2_3/ch_123).

Leszczyna, R. 2004. "Evaluation of Agent Platforms." Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems, 2004. pp1063-1070.

Nikolai, C., and G. Madey. 2009. "Tools of the Trade: A Survey of Various Agent Based Modeling Platforms." Journal of Artificial Societies and Social Simulation, forthcoming.

Powers, D. 2006. "PHP Solutions, Dynamic Web Design Made Easy." New York: Springer-Verlag, Inc.

Railsback, S.; S. Lytinen; S. Jackson. 2006. "Agent-Based Simulation Platforms: Review And Development Recommendations." Simulation 82: 609-623.

Sebesta, R. 2008. "Programming the World Wide Web." Pearson Education, Inc.

Serenko A., and B. Detlor. 2002 "Agent Toolkits: A General Overview of The Market and an Assessment of Instructor Satisfaction with Utilizing Toolkits in the Classroom." Working Paper #455.

Simmers, L. 2003. "Diversified Health Occupations." New York: Delmar Learning, p278.

Tobias, R., and C. Hofmann. 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>.

U.S General Accounting Office. 1992. "Quantitative Data Analysis: An Introduction." GPO Access; Available at <http://archive.gao.gov/t2pbat6/146957.pdf>.

U.S. General Accounting Office. 1993. "Developing and Using Questionnaires," GPO Access; Available at <http://archive.gao.gov/t2pbat4/150366.pdf>.

Walonick, D. 1997. "Survival Statistics." StatPac, Inc, Available at <http://www.statpac.com/statistics-book/index.htm>.

Yahoo User Interface. Available at <http://developer.yahoo.com/yui/>.

## Biography

Cynthia Nikolai began her academic career studying Mathematics and Computer Science as an undergraduate at Villanova University. She earned a commission in the US Air Force and continued graduate studies in Computer Science with the University of Massachusetts, Amherst. She currently is a Ph.D. Candidate in Computer Science at the University of Notre Dame. Cynthia's research interests include emergency management modeling and simulation, agent-based modeling and simulation, operations research, data mining, and virtualization.

Gregory Madey is a Research Professor in the Department of Computer Science and Engineering at the University of Notre Dame, IN, USA. He received the Ph.D. and M.S. degrees in operations research from Case Western Reserve University and the M.S. and B.S degrees in mathematics from Cleveland State University. His research includes topics in emergency management modeling and simulation, agent-based modeling and simulation, distributed systems, web-services and service oriented architectures, web portals for scientific collaboration, web mining, and scientific databases. He is a member of the ACM, AIS, IEEE Computer Society, Informs, and the Society for Computer Simulation.