Competing Idea Diffusion ABM

(CID-ABM)
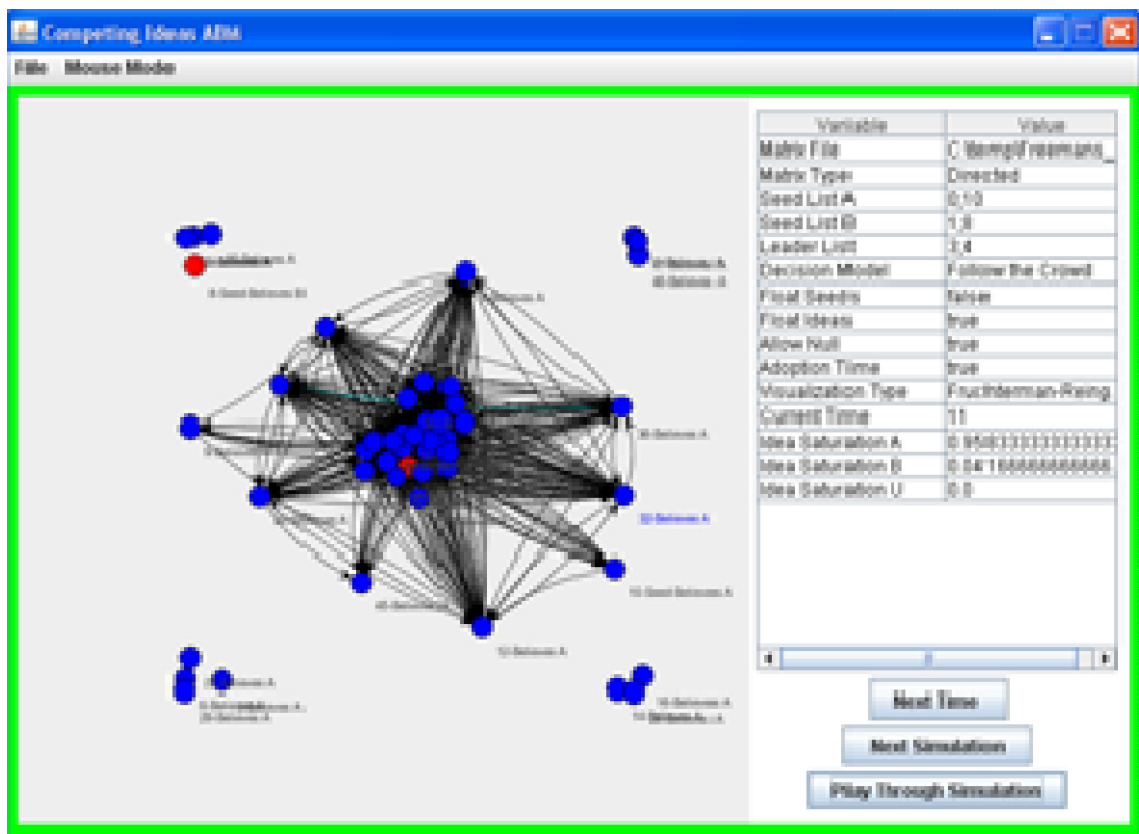
Benjamin Elbirt
September, 2009

I. **Contents**

II. **Software License Agreement**

PLEASE READ THIS SOFTWARE LICENSE AGREEMENT CAREFULLY BEFORE INSTALLING OR USING THE SOFTWARE.

BY OPENING THE PACKAGE, DOWNLOADING THE PRODUCT, OR USING THE EQUIPMENT THAT CONTAINS THIS PRODUCT, YOU ARE CONSENTING TO BE BOUND BY THIS AGREEMENT. IF YOU DO NOT AGREE TO ALL OF THE TERMS OF THIS AGREEMENT, RETURN THE PRODUCT TO THE PLACE OF PURCHASE FOR A FULL REFUND, OR DO NOT DOWNLOAD THE PRODUCT.

Single User License Grant: Benjamin Elbirt ("Owner") grants to Customer ("Customer") a nonexclusive and nontransferable license to use **CID-ABM 1.0©** ("Software") in object code form solely on a single central processing unit owned or leased by Customer.

Multiple-Users License Grant: Benjamin Elbirt ("Owner") grants to Customer ("Customer") a nonexclusive and nontransferable license to use **CID-ABM 1.0©** ("Software") in object code form: (i) installed in a single location on a hard disk or other storage device of up to the number of computers owned or leased by Customer for which Customer has paid a license fee ("Permitted Number of Computers"); or (ii) provided the Software is configured for network use, installed on a single file server for use on a single local area network for either (but not both) of the following purposes: (a) permanent installation onto a hard disk or other storage device of up to the Permitted Number of Computers; or (b) use of the Software over such network, provided the number of computers connected to the server does not exceed the Permitted Number of Computers. Customer may only use the programs contained in the Software (i) for which Customer has paid a license fee (or in the case of an evaluation copy, those programs Customer is authorized to evaluate. Customer grants to Owner or its independent accountants the right to examine its books, records and accounts during Customer's normal business hours to verify compliance with the above provisions. In the event such audit discloses that the Permitted Number of Computers is exceeded, Customer shall promptly pay to Owner the appropriate licensee fee for the additional computers or users. At Owner option, Owner may terminate this license for failure to pay the required license fee.

Customer may make one (1) archival copy of the Software provided Customer affixes to such copy all copyright, confidentiality, and proprietary notices that appear on the original.

EXCEPT AS EXPRESSLY AUTHORIZED ABOVE, CUSTOMER SHALL NOT: COPY, IN WHOLE OR IN PART, SOFTWARE OR DOCUMENTATION; MODIFY THE SOFTWARE; REVERSE COMPILE OR REVERSE ASSEMBLE ALL OR ANY PORTION OF THE SOFTWARE; OR RENT, LEASE, DISTRIBUTE, SELL, OR CREATE DERIVATIVE WORKS OF THE SOFTWARE.

Customer agrees that aspects of the licensed materials, including the specific design and structure of individual programs, constitute trade secrets and/or copyrighted material of Owner. Customer agrees not to disclose, provide, or otherwise make available such trade secrets or copyrighted material in any form to any third party without the prior written consent of Owner. Customer agrees to implement

reasonable security measures to protect such trade secrets and copyrighted material. Title to Software and documentation shall remain solely with Owner.

LIMITED WARRANTY. Owner warrants that for a period of ninety (90) days from the date of shipment from Owner: (i) the media on which the Software is furnished will be free of defects in materials and workmanship under normal use; and (ii) the Software substantially conforms to its published specifications. Except for the foregoing, the Software is provided AS IS. This limited warranty extends only to Customer as the original licensee. Customer's exclusive remedy and the entire liability of Owner under this limited warranty will be, at Owner option, repair, replacement, or refund of the Software if reported (or, upon request, returned) to the party supplying the Software to Customer. In no event does Owner warrant that the Software is error free or that Customer will be able to operate the Software without problems or interruptions.

This warranty does not apply if the software (a) has been altered, except by Owner, (b) has not been installed, operated, repaired, or maintained in accordance with instructions supplied by Owner, (c) has been subjected to abnormal physical or electrical stress, misuse, negligence, or accident, or (d) is used in ultra-hazardous activities.

DISCLAIMER. EXCEPT AS SPECIFIED IN THIS WARRANTY, ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS, AND WARRANTIES INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE, ARE HEREBY EXCLUDED TO THE EXTENT ALLOWED BY APPLICABLE LAW.

IN NO EVENT WILL Owner BE LIABLE FOR ANY LOST REVENUE, PROFIT, OR DATA, OR FOR SPECIAL, INDIRECT, CONSEQUENTIAL, INCIDENTAL, OR PUNITIVE DAMAGES HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY ARISING OUT OF THE USE OF OR INABILITY TO USE THE SOFTWARE EVEN IF Owner HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. In no event shall Owner liability to Customer, whether in contract, tort (including negligence), or otherwise, exceed the price paid by Customer. The foregoing limitations shall apply even if the above-stated warranty fails of its essential purpose. SOME STATES DO NOT ALLOW LIMITATION OR EXCLUSION OF LIABILITY FOR CONSEQUENTIAL OR INCIDENTAL DAMAGES.

The above warranty DOES NOT apply to any beta software, any software made available for testing or demonstration purposes, any temporary software modules or any software for which Owner does not receive a license fee. All such software products are provided AS IS without any warranty whatsoever.

This License is effective until terminated. Customer may terminate this License at any time by destroying all copies of Software including any documentation. This License will terminate immediately without notice from Owner if Customer fails to comply with any provision of this License. Upon termination, Customer must destroy all copies of Software.

Software, including technical data, is subject to U.S. export control laws, including the U.S. Export Administration Act and its associated regulations, and may be subject to export or import regulations in

other countries. Customer agrees to comply strictly with all such regulations and acknowledges that it has the responsibility to obtain licenses to export, re-export, or import Software.

This License shall be governed by and construed in accordance with the laws of the State of New York, United States of America, as if performed wholly within the state and without giving effect to the principles of conflict of law. If any portion hereof is found to be void or unenforceable, the remaining provisions of this License shall remain in full force and effect. This License constitutes the entire License between the parties with respect to the use of the Software.

I. Product Specification

1. Product Overview

CID-ABM 1.0© is a software program written for Windows Operating Systems and allows a researcher to investigate the propagation of competing [information] ideas through a [social network] system with controls for which agents start with the idea [seeds], leadership characteristics, adoption time (Solomon, Weisbuch, et al., 2000), and dynamic idea adoption [floating decisions]. JUNG (Java Universal Network/Graph Framework) is used for graphical output during simulation. Data outputs include time stamped calculations, adoptions and saturation percentages.

2. Software Installation

This software is stand-alone and will run from any location on the computer. The Java Runtime Environment vs. 1.5x or later is required.

3. Running the Software / Memory Expansion

The software is run using the standard java –jar command. Options for memory expansion can be used for larger data sets.

> *java –Xms**1M** –Xmx**2M** –jar cid-abm.jar*

where **1M** is the amount of memory to use for the initial heap size and **2M** is the amount of memory to use for the maximum heap size. The **M** is used to represent megabytes.

> *Java –Xms1000M –Xmx2000M –jar cid-abm.jar*

will launch the application with an initial heap of 1 GB and a maximum heap of 2 GB.

4. Using the Included Short-Cut/BAT File

The zip file includes a BAT file that can be used for creating a desktop shortcut. You should modify the contents of the bat file to represent your system memory capacity to avoid run-errors.

5. Known Issues

JUNG visualization is slow with networks larger than 75 to 100 nodes on a 3 GB 3.4 ghz processor with a 500mb Video card. Although the system will draw networks of 2000 nodes the processing time and animation is slow, choppy and resource intensive.

6. Bug Collection & Reporting

Errors should be reported to the author at elbirt@elbirttechnologies.com.

II.   About the Model

1. Prerequisite Knowledge

This documentation/application assumes the reader has knowledge regarding social networks and the visualization / analysis there of, matrix algebra and agent based modeling.

2. The Research Problem

Two ideas about a given subject can manifest in two isolated locations, but the small-world phenomena shows it takes only a few "hops" along a social network to reach from one end to the other (Milgram, 1967; Watts & Strogatz, 1998). Further, the ideas do not necessarily have to manifest in two isolated groups. Rather, they can form just about anywhere in any number of individuals. Further, Solomon, Weisbuch, et al. (2000) found the time an Agent has the idea impacts the strength of their belief in that idea.

How then do network structures and agent characteristics impact the spread of competing ideas, influence the outcome and impact the rate in which that outcome is achieved?

3. Research Questions

Given a social network of individuals and the existence of two ideas each in one or more individual's possession:

1. How do the ideas spread through the network?
2. What is the final distribution of ideas among the members of the network?
3. How long does it take for the ideas to spread?
4. Do the distributions settle, change in a rhythmic pattern or continuously fluctuate?
5. What influence do leaders have on the spread of ideas?
6. What influence does network structure play on the spread of ideas?
7. Which members of the network have the most influence?
8. Can the information obtained through these questions be used to manipulate the outcome?

CID-ABM© is a software package designed to look at these questions and attempt to find the answers. To do this, a simulation model has been designed that uses a connection matrix to determine how the idea is transmitted. An agent will adopt an idea based on a decision model and the distribution of ideas among the agent's neighbors.

4. Philosophical Applications

The following are example research areas that have network data applicable to this application.

1. The US Legislative process requires co-sponsorship and support of legislation, or bills, by many members of the US Senate to become law. This requires the senators work together even though they are members of different parties and have different philosophies regarding the law and the direction of the US social system. Networks based on senator co-sponsorship of bills has been used to study the way legislature is conducted and the influence of party and other factors on bills that become law (Fowler, 2006a,b).

2. Co-authorship Networks based on journal citation data have been used to study various aspects of academia/science including the features and co-publication of scientific information (Braun, 2001), collaboration (Newman, 2004), evolution (Barabási, Jeong, et al., 2002) and productivity (Kretschmer, 2004).

3. Social networking websites such as Facebook, MySpace and LinkedIn provide details regarding intimate relationships among a group of individuals who actively manage and tighten their networks. There are many areas of research in online social networks including knowledge sharing and influence (Subramani &Rajagopalan, 2003), social capitol (Ellison, Steinfield, & Lampe, 2007) and topology (Ahn, Han, et al., 2007) among many others.

III. Decision Logic

At each point in time, for each Agent (i):

Determine if the Agent (i) can modify its decision based on parameters Floating Decision and Floating Seed and the agent's current idea position (A, B or U).

IF the Agent (i) is able to change its current idea position, determine the distribution of ideas among the Agent's (i) neighbors:

> *FOR EACH* Agent (j) that links to Agent (i) determine the idea distribution as:
>
>> Set adoption value for the idea in possession of Agent (j) as 1.0.
>>
>> *IF* Weighted Edges is Active (true):
>>
>>> Set the adoption value to the edge strength for the edge originating at Agent (j) to Agent (i).
>>
>> *IF* Adoption Time is Active (true):
>>
>>> Add to the adoption value the time since adoption (t-T) where *t* is the current time and *T* is the time of adoption. Thus, the longer the agent has the idea the more influence the idea has.
>>
>> *IF* Agent (j) is a leader:
>>
>>> Apply the Leadership Model adjustment to the idea value.
>
> Add the idea value for each Agent (j) to the idea distribution for the given idea adopted by Agent (j).
>
> Divide the idea value totals by the sum of all idea values to obtain a percentage distribution.

Apply the decision model to determine if the Agent (i) adopts an idea or is undecided given the Allow Null parameter.

At the end of each point in time apply any changes to Agent idea position.

The changes to the idea positions are done at the end of the time period to avoid recursive impact on the decision distributions during a time period. This simulates the notion decisions are made by everyone at once in a given point of time and then everyone re-evaluates before changing their mind.

Diagram 1 – Initial CID Interface

IV. Configuration Interface

The configuration interface provides buttons for [Load]ing saved definitions, [Add]ing files to the configuration list, [Remov]ing files from the list, [Clear]ing the list, [Save]ing the definitions to a file and [Run]ning the simulations.

1. Definitions Explained

Each simulation is represented by a row in the configuration interface. The column header describes the definition content for said column. Some of the columns will provide drop-down lists and/or checkboxes as options. Others are for text input.

- **Matrix Name** – This is the file containing the data to perform the simulation on. Every node in the network must be connected to at least one other node. Relations can be binary or weighted by positive non-zero values. All agents are given numerical id's starting with zero (0).
- **Input Type** – This is the type of Matrix/Network to enter. Options include:
    - *Delimited MX* – This is a tab delimited ASCII matrix with no labels.
    - *MS Excel MX* – This is an excel file containing a matrix in the first spreadsheet. All other spreadsheets are ignored.
    - *Delimited Link Pairs* – This is a tab delimited ASCII file containing link pairs. Each link pair should contain the ID # of the two nodes and optional connection/edge strength delimited by a tab. Missing edge strengths are assumed 1.0.
    - *Jacob's Ladder 12©* - This is an XML file generated with NetXML – a software package that accompanies Jacob's Ladder 12© and creates specific XML files for

Jacob's Ladder 12© input. Only the first matrix specified in the XML file will be used.

- **Matrix Type** – This is the type of matrix regarding edges; Directed or Undirected. The matrix type impacts the visualization.
- **Seed List A / Seed List B** – These are the semicolon (;) delimited lists of agents to seed with the specified idea. Each list must be unique; i.e. members in one list cannot be in the other. Both lists must have at least one agent specified.
- **Leader List** – This is a semicolon (;) delimited list of agents to consider leaders. Leave this field empty if there are no leaders or use the word "All" (case insensitive) to consider all Agents leaders[1].
- **Leader Model** – This is the model to use for determining the leadership effect. Current models are:
  - *Edge Weight Adder (EWA)* – The edge weight will be added to the influence score of the leader's idea.
  - *Support Count Adder* – The number of Agents that support the leader's idea and are connected to by the leader (directed only accepts outward links from the leader) are added to the influence score of the leader's idea.
  - *Support Count Multiplier* – The Support Count Adder function is multiplied by the influence score of the leader's idea instead of added.
  - *Support EWA* – The agents who would be used in the Support Count Adder/Multiplier functions have the edge strength scores (leader to agent) added to the influence score of the leader's idea. Thus, the stronger the edge weight between the leader and those who support the leader's idea the larger the value added to the idea's distribution score.
  - *Square (^2)* – The total score will be squared.
  - *Multiplier (2x)* – The total score will be doubled.
- **Decision Model** – This is the decision model to use for idea adoption. Current models are:
  - *Follow the Leader* – This model simulates majority rule and will assign the idea with the largest distribution to the agent. In the event of a tie the NULL (Undecided) choice is selected even if Allow Null is false.
  - *Root for the Underdog* – This model simulates going against the crowd and will assign the idea with the smallest distribution to the agent. In the event of a tie the NULL (Undecided) choice is selected even if Allow Null is false.
  - *50-20 Rule* – This model simulates a majority rules unless the minority is strong enough to cast doubt notion. The model states that at least 50% (0.50) must support one idea AND no more than 20% (0.20) can support the other. The Agent will become undecided (U) in the event that neither A or B meets the requirement AND Allow Null is selected; otherwise they retain their current idea.
- **Floating Seeds** – This is a true/false option to allow seeds to change their idea. Floating Ideas must be true.

---

[1] This option is intended for Support Count Adder, Multiplier and Support EWA Leader Models.

- **Floating Ideas** – This is a true/false option to allow ideas to change for non-seeded agents.
- **Allow Null** – This is a true/false option to allow null choice. The determination of the null choice is based on the Decision Model.
- **Adoption Time** – This is a true/false option to consider time of idea adoption (Solomon, Weisbuch, et al., 2000).
- **Weighted Edges** – This is a true/false option to consider edge weights as the strength of idea adoption (true) vs. a binary 1.0 value (false).
- **Display Layout[2]** – This is the type of display to use for the visualization. Current options are
    - Fruchterman-Reingold
    - Circle
    - Spring
    - Meyer's ISOM

2. Adding / Removing Definitions

Select as many rows as desired, using the control / select keys on the keyboard for multiple selection, and press the [Remove] button to remove said rows. Use the [Add] button to open a file dialog and select as many files to add as needed. Each file will be added with default settings.

3. Saving / Loading the Definitions

These buttons allow for the saving and loading of definitions. Saved definitions are based on the current Configuration Interface content. Loaded definitions are appended to the existing list allowing for the repeated addition of the same definition file or the addition of multiple unique files.

4. Running the Simulation

The [Run] button launches the visualization interface and hides the configuration interface. Prior to launching the directory selection dialog is presented to determine where the output files should be created. **Cancel the directory specification dialog to simulate without creating output files.**

The output files will be generated 1 per row of configuration. If the file to create already exists it will be appended with an index number based on the next available index preventing over-write. Each output contains a data string series that outlines the simulation as it occurs.

5. Simulation Output File

The output is generated as a KEY:VALUE pair where the VALUE will be a specific set of data elements tab delimited and the KEY determines what the data contains as follows:

- **DEF** – This is the definition entry in the Configuration Interface for the simulation to outline what the file contains.

---

[2] The layouts are a function of the JUNG framework. Reference this documentation for more information regarding the layouts.

- **SAT** – This is the saturation percentage distribution for the given simulation time stamp. It is provided as [Time Stamp(TAB)%A(TAB)%B(TAB)%U].
- **CHK** – This is the outcome of performing the Agent Changeable check. . It is provided as [Time Stamp(TAB)Agent ID(TAB)true/false].
- **DST** – This is the outcome of doing a distribution count for the given CHK based on the agent's relations. . It is provided as [Time Stamp(TAB)Agent ID(TAB)%A(TAB)%B(TAB)%U].
- **CHC** – This is the choice statement based on the given DST and Decision Model. . It is provided as [Time Stamp(TAB)Agent ID(TAB)Selected Index(TAB)Index Percentage].
- **SET** – This the statement of setting the given CHC after all agents are checked. . It is provided as [Time Stamp(TAB)Agent ID(TAB)A/B/U].

Example:

```
CHK:1 2      true
DST:1 2      0.1935      0.2580      0.5483
CHC:1 2      0     0.1935483870967742
SET:1 2      A
SAT:1 0.22077922077922077     0.2077922077922078     0.5714285714285714
```
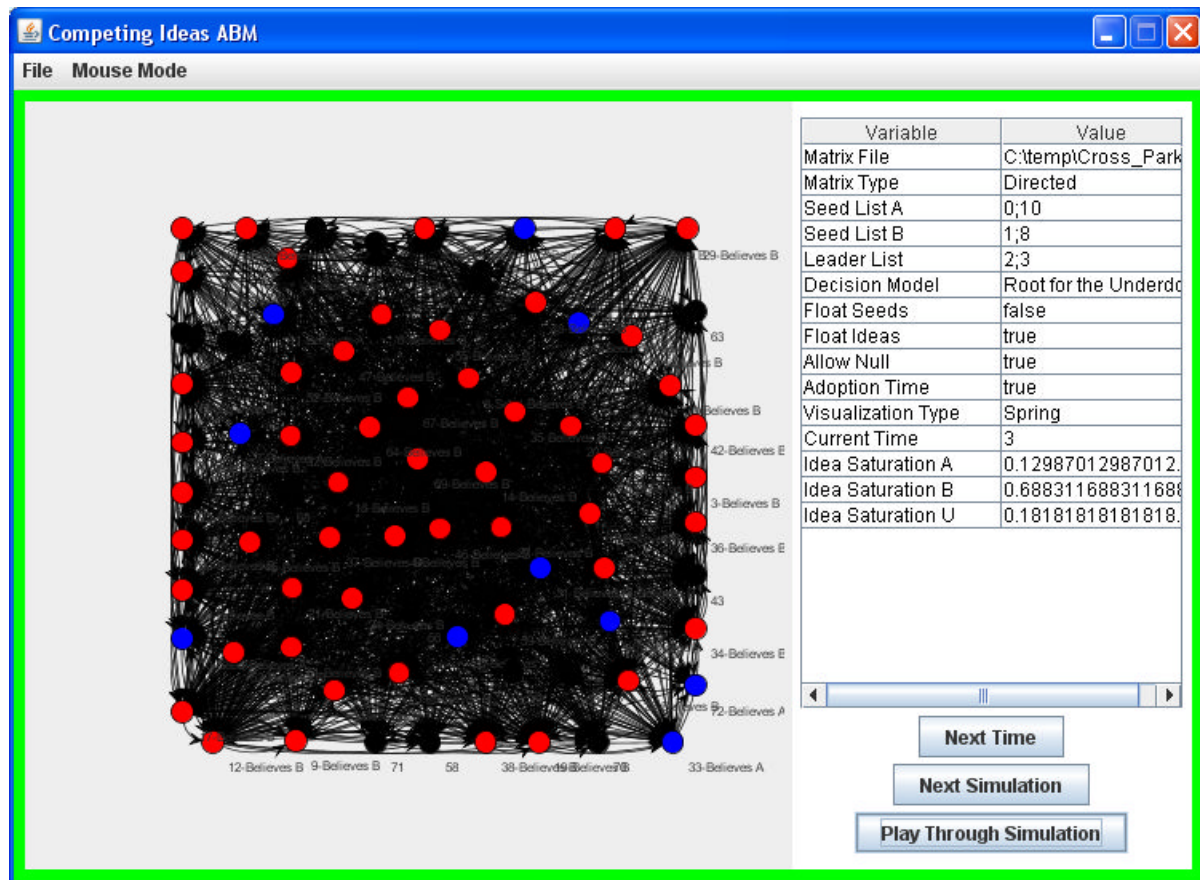
Diagram 2 – Visualization Interface

V. **Visualization Interface**

The visualization interface provides a graph display, data window, and simulation control buttons.

6. Main Menu

The main menu is at the top of the interface and provides **File** and **Mouse Mode** controls.

Mouse modes are transforming and picking; transforming is for scaling and motion of the graph and picking is for moving specific agents in the graph.

The File menu provides the option to exit the application or return to the configuration interface.

7. Graph Display

The left side of the display contains the graph window generated using the JUNG (2009) java framework. This graph will be one of the types defined in the Configuration Interface – Display Type.

Agents are colored Blue for idea A, Red for idea B and Black for undecided (U)/null.

Transforming mouse controls provide the ability to zoom (middle mouse/scroll wheel), pan (left mouse button), rotate (left mouse button with shift key) and spin (left mouse button with control key).

Picking mouse controls allow for the selection of one or more Agents (left mouse button) and the movement of these agents (move mouse while holding left mouse button).

8. Data Window

The data window provides information regarding the simulation including the definition parameters, current time in the simulation and the current saturation of ideas. Additional information about the network, including size, edge count and sum of edge strengths are provided among the outputs.

9. Next Time, Next Simulation and Play Through Simulation Buttons

These buttons control the simulation. The [Next Time] button moves the simulation forward one point in time while the [Play Through Simulation] will continuously advance the simulation until stopped.

The [Next Simulation] button will end the current simulation and start the next one as listed in the configuration interface.

VI. **References**

(2009). Java Universal Network/Graph Framework (JUNG).

Ahn, Y.-Y., S. Han, et al. (2007). Analysis of topological characteristics of huge online social networking services. International World Wide Web Conference, Banff, Alberta, Canada.

Barabási, A. L., H. Jeong, et al. (2002). "Evolution of the Social Network of Scientific Collaborations." Physica A **311**(3-4): 590-614.

Ellison, N. B., Steinfield, C., & Lampe, C. (2007). The benefits of Facebook "friends:" Social capital and college students' use of online social network sites. *Journal of Computer-Mediated Communication*, *12*(4), article 1.

Fowler, J. H. (2006). "Connecting the Congress: A Study of Cosponsorshp Networks." Political Analysis **14**(4): 456-487.

Fowler, J. H. (2006). "Legislative Cosponsorship Networks int he U.S. House and Senate." Social Networks **28**(4): 454-465.

Khan, A. (2009). JExcelApi.

Kretschmer, H. (2004). "Author productivity and geodesic distance in bibliographic co-authorship networks, and visibility on the Web." Scientometrics **60**(3): 409-420.

Milgram, Stanley (1967). "The Small World Problem". Psychology Today 1 (1): 60-67.

Newman, M. E. J. (2004). "Coauthorship Networks and Patterns of Scientific Collaboration." PNAS **101**(1): 5200-5205.

Solomon, S., G. Weisbuch, et al. (2000). "Social Percolation models." Physica A **277**: 239-247.

Subramani, M. R. and B. Rajagopalan (2003). "Knowledge-sharing and influence in online social networks via viral marketing." Communications of the ACM **46**(12): 300-307.

Watts, Duncan J.; Strogatz, Steven H. (June, 1998). "Collective dynamics of 'small-world' networks". Nature **393**: 440-442.