RePast

An Introduction to Agent Based Modeling with Repast

Michael North
north@anl.gov
www.cas.anl.gov

Copyright 2004 Argonne National Laboratory
“Our goal with Repast is to move beyond the representation of agents as discrete, self-contained entities in favor of a view of social actors as permeable, interleaved and mutually defining, with cascading and recombinant motives.”

“We intend to support the modeling of belief systems, agents, organizations and institutions as recursive social constructions”

repast.sourceforge.net
What is ABMS?

- ABMS seeks to create electronic laboratories (“e-laboratories”) that allow experimentation with simulated complex systems:
  - ABMS uses sets of agents and frameworks for simulating the agent’s decisions and interactions
  - ABMS can show how a system could evolve through time in a way that is difficult to predict from knowledge of the behaviors of the individual agents alone
- ABMS focuses on individual behavior with the agent rules are often based on theories of the individual such as Rational Individual Behavior, Bounded Rationality or Satisficing
- Based on these simple types of rules, ABMS can be used to study how patterns emerge
- ABMS may reveal behavioral patterns at a macro (system) level that are not obvious from an examination of the underlying agent rules alone – these patterns are called “emergent behavior”
A Complex Adaptive System (CAS) is made up of agents that interact and reproduce while adapting to a changing environment.

Researchers such as John Holland are trying to isolate fundamental causes of adaptation and emergence of system-wide properties – in any CAS.

John Holland has identified the following properties and mechanisms that are common to all CAS:
- Nonlinearity
- Diversity
- Aggregation
- Flows
- Tagging
- Internal models
- Building blocks

ABMS incorporates some of the properties and mechanisms of CAS.
There Are Many Examples of Systems Comprised of Interacting Individuals

- Economic markets:
  - Producers
  - Distributors
  - Consumers
- Human immune system:
  - Antibodies
  - Bacteria
  - Viruses
- Social Systems:
  - People
  - Factions
  - Countries
- Ecosystems:
  - Species
  - Individuals
  - Hives
  - Flocks
What Useful Information Can ABMS Provide?

- ABMS can help to provide insight into and predictions of agent behaviors
- ABMS can help to anticipate system dynamics, structures, and possible evolutionary paths including suggesting answers to a variety of questions including the following:
  - What agent rules influence emergent behavior and how do they do so?
  - Will some types of agents tend to dominate?
  - Will changes come quickly or slowly?
  - Will some systems always be in a state of turbulence?
- ABMS can be used to help identify disequilibrium situations and their causes
- ABMS can be used to help identify sources of uncertainty in the underlying system
ABMS is Complementary to Traditional Techniques

- **Analytics**: Analytical modeling seeks to develop rigorous, provable statements about systems

- **Statistical Methods**: Statistical modeling specifies how outputs depend on inputs – systems are represented as a “black boxes”

- **Optimization**: Optimization modeling seeks to find optimal solutions relative to well-defined objectives and subject to specific constraints

- **Discrete Event Simulation**: Traditional discrete event simulation modeling represents the inner workings of dynamic processes and moves those representations forward through time at a system level

Copyright 2004 Argonne National Laboratory
Analytical Modeling Seeks to Develop Provable Statements About Systems

• An example is solving a well-posed problem in classical mechanics
• Difficulties:
  – Analytical models usually focus on global descriptions
  – Analytical models of complex systems can be extremely unwieldy
  – Excessively “heroic” assumptions are required to create analytically solvable models of many systems
  – Many systems cannot be analytically solved at all
Statistical Methods Seek Relationships Between Inputs and Outputs

- Output = f(Input₁, Input₂, Input₃, …., Inputₙ)
  where f is a statistically derived relationship

- Difficulties
  - The derived relationships can be brittle
  - Not sensitive to many assumptions or amenable to “what-if” scenarios
  - Insight into underlying causes is often limited
Optimization Modeling Seeks “Best Values”

- Max $x_1$ and $x_2$: 
  $$(1 - e^{(x_1+x_2)}) \quad \text{(Fitness)}$$
- Subject to: 
  $$3 \ x_1 + 5 \ x_2 < 100 \quad \text{(Size constraint)}$$

- Difficulties
  - Optimization models usually focus on global descriptions
  - Even for well defined problems, finding optimal solutions can be extremely difficult
  - Brittle formulations and solution points can result
Discrete Event Simulation Modeling Represents the Detailed Steps in a Process As It Unfolds Over Time

- **Difficulties**
  - The emphasis is on fixed processes instead of adaptive actors
  - Defining the process representation can be difficult since there is no a clear delineation between too little and too much detail

Copyright 2004 Argonne National Laboratory
Where Did ABMS Come From?

40s
- Cellular Automata  Stanislaw Ullam, John von Neumann

70s
- Genetic Algorithms  John Holland: adaptation in natural and artificial systems (60s-70s)
- Game of Life  John Conway (70s)
- Artificial Life  Chris Langton (80s)

80s
- Santa Fe Institute (SFI) for CAS research (84)
- SFI Business Network established (92)
- CAS spin-off companies (Bios Group, Complexica, CASA)

90s
- Artificial Societies  J. Epstein & R. Axtell: Sugarscape (96)
- Small World Networks  Duncan Watts (98)

00s
- Dynamic Social Networks (02)
- Synchronization, Spontaneous Order, Steve Strogatz (03)
• The REcursive Porous Agent Simulation Toolkit (Repast) is a free, open source agent-based modeling and simulation (ABMS) library
• Repast is written entirely in Java and requires version Java 1.4 or greater
• Repast seeks to support the development of extremely flexible models of social agents, but is not limited to social modeling alone
• Users build simulations by incorporating Repast library components into their own programs
• More information on Repast, as well as free downloads, can be found at the Repast development team’s home page, http://repast.sourceforge.net/
• Repast was launched by David Sallach and Nick Collier in 2000
• Since then, the Repast Organization for Architecture and Design (ROAD) was formed to maintain Repast in an open environment:
  – ROAD is a non-profit volunteer organization committed to developing Repast
  – The ROAD Board includes members from a wide range of government, academic and industrial organizations
Repast is the One of the Most Popular Among a Range of Available ABMS Toolkits

Ease of Model Development

High

Low

Easy

Hard

Selected Example ABMS Toolkits

- DIAS www.dis.anl.gov/DIAS/
- IMT flock.cbl.umces.edu/imt
- Swarm www.swarm.org
- Repast repast.sourceforge.net
- Ascape www.brook.edu/es/dynamics/models/ascape
- Object Oriented Languages (Java, C++, etc.)
- Structured Languages (C, Pascal, etc.)
- Mathematics Packages (Mathematica®, etc.)
- Spreadsheets
- NetLogo ccl.northwestern.edu/netlogo/
- StarLogo www.media.mit.edu/starlogo
- Participatory Simulation
The Repast Simulation System has Two Layers

- The *core layer* runs general-purpose simulation code written in Java:
  - This component handles most of the “behind the scenes” details
  - Repast users do not normally need to work with this layer directly
- The *external layer* runs user-specific simulation code written in either Java or Python:
  - This component handles most of the “center stage” work
  - Repast users work with this layer:
    - Basic models can be written in Python using the SimBuilder interface
    - Basic or advanced models can be written in Java
SimBuilder is a Rapid Application Development Environment for Repast Simulations

- SimBuilder users can visually construct simulations out of component pieces and specify the behavior a subset of Python
- SimBuilder is useful for:
  - Learning Repast
  - Rapid prototyping
SimBuilder Simulations are Organized into Projects

- Each project is composed of components chosen from the component palette and displayed in the project pane:
  - Components in the project pane are organized hierarchically such that the project component is the root of the hierarchy
  - Components further up the hierarchy are considered parents to those child components below
  - These components are composed of properties that are customized by the model builder to create the simulation
  - The properties for the current selected component are displayed in the property pane
- SimBuilder compiles the components in the project pane into Java code compatible with the Repast simulation framework
Building a Simple Simulation is Easy!

- Add components from the component palette to the property pane:
  - A model component, such as Default SimBuilder Model
  - A child agent component, such as Default SimBuilder Node
- Customize these components by editing their properties
- Compile the project:
  - The model is responsible for setting up the various pieces of the simulation including the agents
  - The agent component acts as a template for the creation of agents by the model
- Run the resulting Repast simulation:
  - The model creates some number of agents
  - The agents define properties and behavior
- Model verification and validation (V&V) are then needed if the simulation is to be used for an application...
Repast Java Simulations Work the Same Way as SimBuilder Applications
Simple Repast Java Simulations are Created Similarly to SimBuilder Models

- Define the components
  - Define the model:
    - Define the agent world and its properties
    - Define the display surfaces
    - Define the data recorders
    - Define the graphs
  - Define agents:
    - Define the basic agent properties
    - Specify probable agent properties
    - Define the basic agent behaviors
    - Schedule the agent behaviors
- Compile the project
- Run the resulting Repast simulation
- Do V&V…

Copyright 2004 Argonne National Laboratory
Repast Includes a Range of Important Features

- Social network modeling tools are built in
- Geographical information systems (map) modeling tools are built in
- Repast has a concurrent discrete event scheduler…

Copyright 2004 Argonne National Laboratory
Simulations proceed by popping events (or actions as they are called in Repast) off an event queue and executing them:

- These actions are such things as “move all agents one cell to the left”, “form a link with your neighbor’s neighbor” or “update the display window”
- The model developer determines the order in which these actions execute relative to each other using ticks

A tick acts as a temporal index for the execution of actions:
- For example, if event X is scheduled for tick 3, event Y for tick 4, and event Z for tick 5, then event Y will execute after event X and before event Z

- Actions scheduled for execution at the same tick will be executed with a simulated concurrency

In this way, the progression of time in a simulation can be seen as an increase in the tick count.
• The introduction of actions with duration introduces flexibility in the scheduling of event
• Events can now be naturally modeled as background actions
• Actions that can:
  – Be started and run in the background for some period of time
  – Run concurrently with other actions with compatible tick counts
  – Block the execution of other actions with higher tick counts until the current action is completed
• For example:
  – A process to be modeled contains some long-running and complicated calculation that can be started at time t with results needed at t + 5
  – There are actions that can be run concurrently over time t to t + 5
  – Then we can model this calculation as an action with duration
  – In terms of implementation, this action will run in its own thread that is amenable to being run on a separate processor or even on another machine
The Are Many Examples
Included in the Repast Distribution

Heat Bugs
Hexabugs
Sugar Scape
Jain
Jin-Gir
Shelling GIS

Copyright 2004 Argonne National Laboratory
Now for Some Fun!

- Let’s do some hands-on exercises…
Are there additional questions?

Michael North
north@anl.gov
www.cas.anl.gov