SwarmFest 2001 Santa Fe

www.swarm.org

Fifth annual meeting of the Swarm user group

Hotel Santa Fe
Santa Fe New Mexico

April 28 - 30, 2001
Sponsored by: Swarm Development Group
http://www.swarm.org

Program Schedule

Please note: this schedule is not 100% final and is subject to change.

Saturday April 28

6:00 pm	Reception/Dinner/Party at the Santa Fe Institute, with *FreshUp. Transport will be provided from the Hotel Santa Fe to the Institute.*

Sunday April 29

8:00 am	**Meeting starts [breakfast pastries, coffee, etc.]**
8:30am	Introduction, Marcus Daniels, Swarm Development Group
9:00am	Chris Langton *Swarm Models of Cooperation in Traditional Agricultural Systems*
9:45am	Michael North *Agent-Based Modeling of Electricity Markets*
10:30am	**Break [coffee, soft drinks]**
10:45am	Pietro Terna *JVEFrame: a Virtual Enterprise Frame in Swarm*
11:30am	Glen Ropella *Software Engineering Considerations for Agent-based Models*
12:15pm	**Lunch [your own arrangements]**
1:30pm	Gary An *Using Agent Based Computer Simulation to Characterize the Inflammatory Response*
2:15pm	Xiaodong Li *Investigation on Critical Density in a Fire Spread Model using a Multi-agent Approach*
3:00pm	**Break [coffee, soft drinks]**
3:15pm	Jim Anderson *An Agent-Based Model of Fish Behavior: Part III*
4:00pm	Rich Colbaugh *An Agent-Based Approach for Modeling and Simulation of Hybrid Dynamical Systems*
4:45pm	Michael Krumpus *The Evo Artificial Life Framework*
6:00 pm	**Dinner [Hotel Santa Fe]**

Monday April 30

8:00am	**Day begins [breakfast pastries, coffee, etc.]**
8:30am	Ferdinando Villa *Integrating Modelling Architecture*
9:30am	Jorge Goic *Distributed Simulations using Swarm*
10:30am	**Break [coffee, soft drinks]**
10:45am	Xiaodong Li *A Parallel Genetic Algorithm Implemented in Swarm*
11:15pm	Steve Jackson *Implementing Multiple Model Swarms*
11:30pm	Alessandro Perrone *VSB: Building a Full Swarm Simulation*
12:00pm	**Lunch [your own arrangements]**
1:15pm	Steve Railsback *Publishing Agent-based Research*
2:15pm	Jay Palmer *An Efficiency Measure for Agent-Based Modeling*
3:00pm	**Break [coffee, soft drinks]**
3:15pm	Aaron Fritz *An Independent Verification of the Clarke Urban Growth Model*
4:00pm	Doug Donalson *Modeling in Continuous Space and Time*
4:45pm	Wrap up discussion

Talk abstracts

Panels

Publishing Agent-based Research Steve Railsback <lra@northcoast.com> (Lang, Railsback & Associates)

Publication in the mainstream scientific literature is essential for the credibility and dispersal of our research. Publication of agent-based simulation often presents special challenges. Some of these challenges are because agent-based simulation is a new approach: convincing editors and reviewers that research results are of general applicability, overcoming objections to models with too many parameters, justifying selection of methods and inputs, satisfying the need for "validation". Other challenges are more technical: how to adequately document a model within a journal article's size limitations, how to present results derived from graphical interfaces. In this panel discussion, researchers who have attempted and sometimes succeeded in publishing agent-based research will identify such challenges and ways in which they can be overcome or circumvented. We hope to include the perspectives and advice of journal editors who have managed the review of agent-based papers.

Modelling talks

Using Agent Based Computer Simulation to Characterize the Inflammatory Response Gary An <Docgca@aol.com> (Cook County Hospital, Department of Trauma)

The non-adaptive inflammatory response is a complex system that maintains the human body against injury. However, severe perturbations to this system may lead to pathologic conditions in which the system behaves in paradoxical fashion. The clinical manifestation of this is Systemic Inflammatory Response Syndrome (SIRS). The management of SIRS represents the greatest challenge in critical care medicine. Although many of the mediators of this process have been characterized, attempts to manipulate the course of the disease using this knowledge have been unsuccessful, even detrimental. There is a gap between the physiologic manifestations of SIRS/MOF and the underlying cellular/molecular pathophysiology. We propose that clinically apparent organ-level physiology represents emergent properties of the underlying cellular/molecular mechanisms. A failure to account for the nonlinear, complex nature of the inflammatory response is a failure to recognize the emergent basis of the physiologic organ response, and is the root of the difficulty in formulating effective treatment regimes.

Agent-Based Computer Simulation (ABCS) may be a means to characterize the dynamics of the inflammatory response. Using the known cellular and molecular mediators of the process as agents it may be possible to construct models that would maintain the complexity of the inflammatory response. Not only does the current extremely abstract preliminary model demonstrate the pattern of dynamics seen in the inflammatory response, it can also be shown to qualitatively reproduce the dynamics of existing clinical trials. It is proposed that pattern oriented analysis could be used as a means of validation for these models. Eventually it is hoped that more sophisticated models will be used to generate state-space maps of the inflammatory response, enabling identification of phase transitions between the "normal", dynamically stable state and its unstable, pathologic state (SIRS). ABCS is envisioned as an engineering tool in designing treatment regimes and as an aid in the understanding of how the various components of the system interact to produce organ-level physiology.

Some of the information in this talk was presented at the 3rd ICCS from last year, and the data from the simulation was presented as a poster at this years Annual Meeting of the Society of Critical Care Medicine.

An Agent-Based Model of Fish Behavior: Part III Jim Anderson <jim@cbr.washington.edu> (University of Washington, Columbia Basin Research) and Nick Beer <nick@cbr.washington.edu> (University of Washington, Columbia Basin Research)

This is the third round of modeling the migration of juvenile salmon through the Columbia River with agent-based models. In the first round, reported at SwarmFest 2000[1], we successfully modeled the hydrological and topographical features of Little Goose Reservoir in the Snake River. Using a simple algorithm, where fish swam with a downstream velocity plus a random velocity dependent on habitat preference, we were able to fit observations of fish travel time and survival through the reservoir. However, when we altered hydrological conditions to represent the reservoir drawdown proposed as a salmon recovery action, the model failed spectacularly. The little smolts were speed demons swimming at many times the water velocity. In the second round, presented at the Ecological Society Annual meeting in August 2000, we made the fish a little smarter with their movements depending on a memory factor that altered their motivations to swim, feed and avoid predators. The model produced little black dots that moved around the computer screen eating little pink dots while avoiding big red ones. It had some elements representing how biologists think fish think (!) but it lacked a coherent structure and set of unifying principles. As a consequence, it was difficult to relate the dots to observations of fish in nature or controlled experiments.

Hoping the third time's a charm, we started from scratch, defining fish interactions in a game-theoretic approach where the choice of behavior is defined by the expected utility of the behavior, which in turn is defined by the intrinsic utility and the probability of obtaining the it. All pretty standard game-theory stuff so far: which behavior is selected at any moment depends on its utility. It became interesting though, when we devised a formal structure of agents and associated behaviors. The agents a fish encounters are prey, predator, boundary and habitat. Behaviors are defined in terms of swimming speed and direction. For a fish/agent interaction we define two classes of behavior: tactical when the fish is in contact with an agent and strategic when it is not. Tactical behaviors optimize the encounter outcome and strategic behaviors alter the chance of future encounters. Tactical probabilities are defined in terms of fish distances to the agents, while strategic probabilities are defined in terms of the history of tactical probabilities as weighted by fish memory coefficients. Utilities have one part dependent on the agent and another part dependent on the fish. Laboratory experiments on predator-prey interactions and patch selection, interpreted within this structure, provide a plethora of data with which to calibrate our model. This approach simplifies how we characterize behavior and it is our hope that it is extensible to larger more complex systems.

An Independent Verification of the Clarke Urban Growth Model : Are the Cellular Automatic Transition Rules Working Alone?
Aaron Fritz <afritz@cc.usu.edu> (Utah State University)

Original attempts at creating large-scale urban models (LSUMs) met with much failure in the 1960s. Repeated attempts at replicating the results of many early LSUMs often failed, creating questions over their reliability. With advances in computer technology and the development of new urban growth theories, however, some researchers feel that the problems associated with the early LSUMs are no longer relevant. A resurgence has thus occurred in the development of LSUMs over the last decade.

I will be describing a proposed research project where I will attempt to verify the results of a recent LSUM, the Clarke Urban Growth Model (UGM). The Clarke UGM is a cellular automaton simulation (CA) written in the C-programming language under the direction of Dr. Keith Clarke at the University of California at Santa Barbara (UCSB). I will explain how I plan to use the Swarm simulation software and the Swarm/Kenge GIS library in order to build a test simulation in which to implement Clarke's urban growth cellular transition rules and what I hope to find by doing so.

Swarm Models of Cooperation in Traditional Agricultural Systems Chris Langton <cgl@swarm.org> (Swarm Development Group)

Stephen Lansing, an anthropologist at the University of Arizona, studies the emergence of cooperation in traditional agricultural systems. Several years ago, he and a colleague built a simple agent-based model of rice agriculture in Bali that proved quite useful for understanding the nature of cooperation among Balinese rice farmers.

Stephen recently received a 3-year $1,000,000 dollar grant from the NSF under their new "Biocomplexity" program to continue and substantially enhance his early work on Bali, with special focus on extending and generalizing the agent-based modeling techniques employed in his early model to facilitate the study of other agricultural traditions. Stephen has chosen to use Swarm as the basis for this extended modeling capability, and I am working with Stephen and several of his colleagues to implement the Swarm framework for this project. I will describe this project and discuss the way we are approaching it via Swarm.

Investigation on Critical Density in a Fire Spread Model using a Multi-agent Approach Xiaodong Li <`Xiaodong.Li@infotech.monash.` (RMIT University, Department of Computer Science)

Small changes in spatial pattern on a landscape can sometimes produce dramatic ecological responses. Such transition ranges are associated with critical environmental conditions such as tree density. As the landscape becomes dissected into smaller patches of trees, landscape connectivity may suddenly become disrupted, which may have important consequences for the behaviors of forest fire, i.e., how it spreads. Landscape connectivity depends not only on the tree density but also many other environmental conditions such as land height, flammability, and wind conditions. To determine how the critical densities are affected by the changes in various landscape conditions, we developed a bushfire spread simulation model using Swarm, a multi-agent simulation framework developed at Santa Fe institute. The model takes into account some of the most influential factors contributing to the development and spread of bushfires. These environmental factors include bush density, bush flammability, heat conditions, land height, wind direction and wind magnitude. With different initial conditions, the fire will either rapidly spread out of control, die prematurely, or survive for moderate periods of time before being overcome by unfavorable conditions. By varying these variables under controlled conditions, this research aims to show how they influence the spread and growth of a bushfire. The paper will first describe the fire simulation model from a multi-agent perspective and the environmental variables that are adopted and implemented in the simulation model. Some experimental results and analysis will then be presented, followed by a summary and directions for future research.

Agent-Based Modeling of Electricity Markets Michael North <`north@anl.gov`> (Argonne National Laboratory, Decision and Information Sciences, Complex Adaptive Systems)

"It's not a pretty picture. The fear has been that people have put on rose-colored glasses." Should the forecast [electric power] shortages materialize, "we would cut off firm load and it would affect thousands of people," Lisa Szot, a spokesperson for California's independent electric power system operator, quoted in the Oil and Gas Journal Online, February 8, 2001.

Complex Adaptive Systems (CAS) Agent-Based Models (ABM) can be applied to investigate multifaceted systems such as electricity markets. As demonstrated recently in California, these markets may fail to meet consumer expectations. CAS ABMs are being developed by Argonne National Laboratory to help provide insights into the types of problems being faced by places such as California. Recent progress in the development of these ABMs will be discussed.

JVEFrame: a Virtual Enterprise Frame in Swarm Pietro Terna <`pietro.terna@unito.it`> (University of Torino, Italy, Department of Economics and Finance)

A model to simulate the effects of change inside enterprises and the role of knowledge in this kind of economic organizations.

jVEFrame (Java Virtual Enterprise Frame, written in Swarm) is built to simulate the effects of deep modifications that may occur in the life of a firm - for example, the adoption of B2B and B2C strategies - and to investigate the role of knowledge into this kind of economic organizations. The jVEFrame structure is highly decentralized: the "orders", i.e. electronic forms (objects in Swarm contest), contain all the information needed for production. There is one order for each product to be done; it reports the production recipes, the performed phases, the cost accounting etc. The production units that execute the orders may be a part of the simulated enterprise or even be structured as external bodies. The model can also be used to evaluate the relations between the adopted information management systems (i.e., P2P knowledge management) and the behavior of the enterprises.

In a more theoretical perspective, the model represents a step toward the simulation of the interactions among and within enterprises, to search for a better definition of entrepreneurship, in the Austrian economic sense of discovering new fields of innovation.

Issues and techniques in agent simulation

An Agent-Based Approach for Modeling and Simulation of Hybrid Dynamical Systems Rich Colbaugh, <`colbaugh@nmt.edu`> (Institute for Complex Additive Systems Analysis) Matt Gaston, <`mgasto1@umbc.edu`> and Kristin Glass <`kglass@nmt.edu`>

The Complex Additive Systems Analysis (CASA) process has demonstrated that a Hybrid Dynamical Systems (HDS) modeling framework provides a rigorous and quantitative methodology for modeling and analyzing complex infrastructure systems. The development of HDS models in a variety of problem domains, coupled with recent developments in agent-based simulation leads to the development of formalized methods for incorporating agents and agent behavior into HDS simulations. This HDS/agent-based approach permits the rigor of the HDS mathematical representation to be retained while enhancing the accessibility of developing agent-based simulations. Along with a brief overview of the CASA process and Hybrid Dynamical Systems, this paper describes a rigorous framework for HDS modeling and simulation from an agent-based simulation perspective. Methods and formalisms for including agent behavior in both the discrete and continuous time components of the HDS are discussed. Additionally, examples will be presented for a financial market, a social system, and a power grid.

Modeling in Continuous Space and Time : When Being Discrete just isn't enough Doug Donalson <`donalson@lifesci.ucsb.edu`> (Cal State LA, Center for Environmental Analysis, Department of Biology)

Most individual based models are constructed using a discrete time/discrete space engine. While, for many applications, this causes no model artifacts, there can be effects introduced by the fine grain quantization of time and space. In discrete time model, the artifacts can occur two ways. The first is from the assumption that the average behavior during a time interval is equivalent to the starting value of the state variables. The second can occur when the different mechanisms that cause state changes within the time interval are not executed randomly. One potential discrete space artifact I call implicit competition. An individual that colonizes a cell, no matter its size, immediately excludes all other individuals from the entire area of the cell.

An alternate method for handling space and time in individual based models is the use of continuous time and space. A continuous time engine is also called an event driven schedule and works from the assumption that no two events (or state changes) ever occur simultaneously. Because of this, time can be advanced state change by state change. In a continuous space model, an individual can occupy any coordinate within the spatial arena, as opposed to being constrained to the center of a cell. I give an overview of implementations of a continuous time/continuous space model vs. a discrete time/discrete space model. I then discuss some of the advantages and disadvantages of each type of model.

Implementing Multiple Model Swarms Steve Jackson <`jackson3@humbolt1.com`> (Jackson Scientific Computing) and Steve Railsback <`lra@northcoast.com`> (Lang, Railsback & Associates)

For a chinook salmon model, we implemented four different model swarms, each with its own space and (in one case) time step. Each model represented one life stage of the salmon. As a fish moves out of one life stage, the fish object is dropped after its instance variables are used to instantiate a new fish object in the next life stage model. An alternative approach in which the fish objects are persistent and "know" which model they belong to has been evaluated in a test code.

An Efficiency Measure for Agent-Based Modeling Jay Palmer <`jpalmer@etl.noaa.gov`> (National Oceanic & Atmospheric Administration, Environmental Technology Laboratory)

Agent based modeling is in need of a quantitative optimization criterion to help remove the arbitrariness in selections of model parameters. This talk proposes the use of an efficiency measure that I call "algorithmic efficiency" for establishing this needed optimization criterion. Algorithmic efficiency is defined as the mutual information between model-predicted patterns and observed patterns divided by the computational memory used by the model. This efficiency has been proven to be bounded between zero and one as efficiencies ought to be. Because this efficiency is based on a statistic (mutual information), the optimizations will be founded on the statistical disciplines required of complex systems modeling. Identifying the computational memory as the fundamental resource is particularly appropriate for memory intensive agent-based models. However, that does not preclude the joint use of other optimization criteria through the use of utility functions involving the algorithmic efficiency, Lagrange multipliers, and selected constraints such as computational speed.

Software Engineering Considerations for Agent-based Models Glen Ropella, <`gepr@swarm.org`> (Swarm Development Group) Steve Railsback, <`lra@northcoast.com`> (Lang, Railsback & Associates) and Steve Jackson <`jackson3@humbolt1.com`> (Jackson Scientific Computing)

Software design is much more important for agent-based models (ABMs) than it is for conventional models, for three reasons. (1) The results of an ABM are the emergent properties of a system of interacting agents that exist only in the software; unlike analytical model results, an ABM's outcomes can be reproduced only by exactly reproducing its software implementation. (2) Outcomes of an ABM are expected to be complex and novel, making software errors difficult to identify. (3) An ABM needs 'systems software' that manages populations of multiple kinds of agents, often has nonlinear and multi-threaded process control, and simulates a wide range of physical and biological processes. We provide seven general software guidelines for complex models are especially important for ABMs. We also provide four additional guidelines specific to ABMs. Strategies for meeting these guidelines include planning adequate resources for software development and using tools like Swarm that are designed specifically for ABMs.

Tools: frameworks and extensions

Browser-based simulation Marcus Daniels <`mgd@swarm.org`> (Swarm Development Group)

I will demonstrate and discuss the Swarm Development Group's new work to develop an internet-aware collaborative graphical agent-based modeling environment. The core work in this effort is technology that allows the the Swarm simulation engine to interoperate with the Mozilla web browser.

Distributed Simulations using Swarm Jorge Goic <`jgoic@erim.org`> (ERIM)

The ability to run distributed simulations is useful in at least two respects. First, it allows a modeler to partition a complex system into manageable components that can then be run on separate machines. This is particularly useful with agent systems in which the agents have heavy computational loads and inter-agent communication is relatively low. A distributed version has the potential for providing increased performance over running the model on a single CPU. Second, if the synchronization architecture allows it, it would be possible to have Swarm models interact transparently with foreign simulation objects. Imagine using Swarm in conjunction with your second favorite simulation tool that may do a better job at describing special aspects of your model. For example, the control algorithms for swarming robotic vehicles could be modeled in Swarm, while their kinematics could be modeled in a kinematic simulator such as RobCAD(tm).

We propose a simple object-oriented architecture for running distributed simulations with Swarm. The architecture is based on the Adapter or Proxy pattern and the synchronization algorithm, called Syncer. Syncer has been successfully applied to other simulation tools in the past linking a Flexis control modeling tool with RobCAD and Siman/Cinema, a discrete event simulator. Syncer is a low-overhead method designed to maximize parallelism in the execution of the distributed components while remaining transparent to the end user. For Swarm we have re-implemented Syncer in Java and use Java RMI or CORBA as the remote component communications framework.

The Evo Artificial Life Framework Michael Krumpus <`krumpus@omicrongroup.org`> (The Omicron Group)

I will present the core concepts behind the Evo Artificial Life Framework and walk through the process of developing an Evo simulation. I will also demonstrate some example simulations. For more info on Evo, see the Evo website[2].

A Parallel Genetic Algorithm Implemented in Swarm Xiaodong Li <`Xiaodong.Li@infotech.monash.edu.au`> (RMIT University, Department of Computer Science)

In this project, we developed a Parallel Genetic Algorithm (or spatially distributed GA) based on Breeder, a simple genetic algorithm developed by J.J. Merelo. This Swarm implementation of PGA takes advantage of the existing Swarm Space Library, which contains classes that can be used for building objects representing a spatial environment that a parallel GA requires. Interactions such as crossover and mutation among the spatially distributed individuals can be scheduled using Swarm's Activity Library. Since Swarm provides many visualization features, the PGA is capable of visualizing dynamically the performance of an evolving GA population with plotted graphs on model parameter values in real time. This implementation also allows modification of some model parameter values during an optimization run, therefore offers advantages over many existing PGA implementations. Two optimization examples will be used to demonstrate the features offered by this specific PGA implementation.

VSB: Building a Full Swarm Simulation : in a single session, without caffeine. Alessandro Perrone <`alex@unive.it`> (Universita' ca' Foscari De Venezia)

VSB represents one of the most straightforward and most rewarding ways to program the Simulations using Swarm Libraries. With very little effort, you can be right up to speed developing your next great simulation, without all that tedious mucking about worrying about the standard Swarm code.

VSB is fully Carbon compliant, allowing you to create simulations from a single source base for all versions of the MacOS from 7.0 through to OS X. The package is aimed squarely at beginners to Swarm Simulations programming, and is ideal for small to medium-sized projects (even of there's no limit to the quantity of objects you can insert in your project). It generates full open source code, so all programmers should find something useful in it.

VSB requires a compiler to compile the sources it generates. You can use every platform in which there's already compiled the Swarm libraries. For further information, please contact the author or visit the Swarm web pages. VSB has been compiled with Metrowerks CodeWarrior version 6.

Integrating Modelling Architecture : Towards a generalized ontology model for integrated, network-enabled, cross-paradigm modelling. Ferdinando Villa <`villa@cbl.umces.edu`> (University of Maryland)

Independent, uncoordinated development of paradigm-neutral, interoperable model components can be made possible by adopting a consistent description framework based on separating the descriptions of general world aspects such as space and time (domains) from their use in the agents' behavior. An extended view of the agent paradigm can encompass most of the modelling activity being done in different disciplines, and conceptual consistency between data and models allows common treatment of structural and functional descriptions. Formal specification of paradigm-specific grammars ensures the extensibility of paradigm-independent data and model repositories and allows pre-validation of model content. After a general introduction, I will present example multi-paradigm models using the prototype Integrating Modelling Toolkit (IMT), the first implementation of the Integrating Modelling Architecture (IMA[3]). The IMT allows network- and web-based, cross-paradigm modelling using the Document Object Model as a generic representational framework, and XML as the technical medium for storing and communicating interoperable model descriptions.

Notes

1. http://www.cbr.washington.edu/papers/swarmfest.html
2. http://www.omicrongroup.org/evo
3. http://ima.umces.edu