SEVENTH ANNUAL SWARM USERS/ RESEARCHERS CONFERENCE

SwarmFest 2003

April 13-15, 2003
McKenna Hall
University of Notre Dame
Notre Dame, IN

Sponsored by:
University of Notre Dame
Department of Computer Science and Engineering
Center for the Study of Biocomplexity
and the
Swarm Development Group (SDG)

Seventh Annual Swarm Researchers Meeting

SwarmFest 2003
McKenna Hall
Notre Dame, IN

Conference Program Schedule

Sunday, April 13 2003

1:00 p.m. - 7:00 p.m.
Pre-Conference Swarm Tutorial, Paul Box, Utah State University and Glenn Ropella, Swarm Development Group

7:00 p.m. - 10:00 p.m.
Poster Session and Opening Conference Reception

Monday, April 14, 2003

8:30 a.m. - 9:00 a.m.
Introduction and Announcements

9:00 a.m. - 10:00 a.m.
Keynote: John Holland, Challenges for Agent-based Modeling

10:00 a.m. - 10:30 a.m.
Break

10:30 a.m. - 11:00 a.m.
Getting Started with an Agent-Based Model, presented by Steve Railsback, Humboldt State University

11:00 a.m. - 11:30 a.m.
Serialization and Stability Analysis in Swarm Simulations, presented by Paul Johnson, University of Kansas

11:30 a.m. - 12:00 p.m.
Agent-based Simulation System of Juvenile Fish Dam Passage, presented by R. Andrew Goodwin, Cornell University and US Army Engineer Research & Development Center

12:00 p.m. - 1:00 p.m.
Lunch — Conference Center Dining Room

1:00 p.m. - 1:20 p.m.
Exploring the Macroscopic Properties of Complex Systems, presented by Paul Harten, U.S. Environmental Protection Agency

1:20 p.m. - 1:40 p.m.
Running a C++ Model under the Swarm Environment, presented by Richard Loew, University of New South Wales, Sydney, Australia

1:40 p.m. - 2:00 p.m.
Novel Technique for PID Tuning by Particle Swarm Optimization, presented by S. Easter Selvan, Karunya Institute of Technology, India

2:00 p.m. - 2:30 p.m.
Why the Navy Needs TSUNAMI, presented by Michael North, Argonne National Laboratory

2:30 p.m. - 3:00 p.m.
Genetic Algorithms in Swarm Modeling, presented by Gianluigi Ferraris, University of Turin, Italy

3:00 p.m. - 3:30 p.m.
Break

3:30 p.m. - 4:00 p.m.
Multi-agent Virtual Histories of the Development of Differentiated Economic Strategies, presented by Darold Higa, University of Southern California

4:00 p.m. - 4:30 p.m.
Complex Motives in Agent Simulation: Proximity Attractors and Their Design, presented by David Sallach, University of Chicago

4:30 p.m. - 5:00 p.m.
The JAS (Java Agent-based Simulation) Library, presented by Michele Sonnessa, University of Turin, Italy

5:00 p.m. - 5:30 p.m.
SOME: An Agent-based Model of Suburban Sprawl, presented by William Rand, University of Michigan

5:30 p.m. - 6:00 p.m.
Swarm Attack to the Busy Beaver Problem, presented by Alessandro Perrone, University of Venice, Italy, and Gianluigi Ferraris, University of Turin, Italy

6:30 p.m.
Reception and Dinner — Conference Center Dining Room

Tuesday, April 15, 2003

8:00 a.m. - 8:30 a.m.
VSB and SimBuilder - An Analysis of Visual Agent Based Tools, presented by Alessandro Perrone, University of Venice, Italy

8:30 a.m. - 9:00 a.m.
Modeling a Swarm of Search and Rescue Robots using StarLogo, presented by Daniel Stormont, Utah State University

9:00 a.m. - 9:30 a.m.
Decision Making and Enterprise Simulation with jES and Swarm, presented by Pietro Terna, University of Turin, Italy

9:30 a.m. - 10:00 a.m.
On the Role of Affect in the Evolution of Social Control in Multi-Agent Societies, presented by Matthias Scheutz, University of Notre Dame

10:00 a.m. - 10:15 a.m.
Break

10:15 a.m. - 10:45 a.m.
Penelope Project: Web-based Textile Production Planning, presented by Matteo Morini, University of Turin, Italy

10:45 a.m. - 11:15 a.m.
Relational Agent Models - a Framework for Modular and Topology-Independent Simulations, presented by Laszlo Gulyas, Hungarian Academy of Sciences, Hungary

11:15 a.m. - 11:30 a.m.
RePast 2.0: Major Changes and New Features, presented by Thomas Howe, University of Chicago

11:30 a.m. - 12:00 p.m.
Complex Dynamics of Simple Stock Market Behavior, presented by Alessandro Perrone, University of Venice, Italy

12:00 p.m. - 1:00 p.m.
Lunch — on your own

1:00 p.m. - 1:30 p.m.
Network Structures, Public Opinion and the Ising Model, presented by Fabio Rojas, University of Chicago

1:30 p.m. - 2:00 p.m.
EcoLab: Agent Based Modeling for C++ Programmers, presented by Richard Lowe, University of New South Wales, Australia

2:00 p.m. - 2:30 p.m.
H-MAS: A Heterogeneous, Mobile, Ad-hoc Sensor-Network Simulation Environment, presented by Bren Mochocki, University of Notre Dame

2:30 p.m. - 3:00 p.m.
Agent-based Modeling of Human and Natural Systems and their Interactions, presented by Saqib Khalil, University of North Texas

3:00 p.m. - 3:15 p.m.
Break

3:15 p.m. - 4:00 p.m.
Conference Wrap-up and Feedback

4:00 p.m.
End of Conference

Seventh Annual Swarm Researchers Meeting

SwarmFest 2003

McKenna Hall

Notre Dame, IN

Abstracts listed in alphabetic order by author's name

Presentation Abstracts

(Presenting author's name in **bold**)

Genetic Algorithms in Swarm Modeling

Implementation and usage topics: samples and highlights.

Gianluigi Ferraris — glf313@libero.it
University of Turin

Starting from John Holland's work, the genetic paradigm has been applied in swarm models devoted to investigate a wide range of problems. GAs may be they selves implemented as a swarm model: single genomas could be individuals that evolve to reach best fitting, performing in a defined environment. The GA could help the user in finding good solutions even if the enormous number of possible strategies makes the problem impossible to handle, using the traditional search methods.

The described approach has been applied to several different problems (optimizing spinning and weaving processes, simulations in economics, turing machines etc.) obtaining good results. Focusing these experiences, the topics to be handled, in order to reach high performances, seem to be:
i) mapping strategies into genomas in a useful way,
ii) avoiding production of non applicable strategies instead of eliminate them,
iii) computing and using strategies evaluation (fitness value) as well,
iv) correct setting of GA's parameters (number of genomas, crossover and mutation rate, etc.),
v) empower the method using, for instance, diploid genomas, clustered and multiple GAs and so on.

Swarm Attack to the Busy Beaver Problem

Gianluigi Ferraris[1] and **Alessandro Perrone**[2]
1) University of Turin
2) University of Venice
e-mail for Ferraris: gf313@libero.it
Perrone: alex@unive.it

Since T. Rado in 1962 defined the busy beaver game, several approaches have used computer technology to search best candidates to solve the Busy Beaver Game. In this paper we follow an "evolutionary approach" to solve it using agent based techniques. This approach includes techniques to reduce the number of inspected Turing machines, to accelerate simulation of Turing machines using Agent Based Techniques, and in particular we use the "Swarm simulation toolkit" to write the package. Our approach uses a variety of learning techniques such as Genetic Algorithms (GA), Classifier Systems, Multiple GA, Random search to explore the universe of the "best solution" to the Game.

Agent-Based Simulation System of Juvenile Fish Dam Passage

R. Andrew Goodwin[1] — Andrew.R.Goodwin@nwp01@usace.army.mil
James J. Anderson[2] — jim@cbr.washington.edu
John M. Nestler[3]

To facilitate the selection of improved fish bypass design measures at Lower Granite Dam on the Snake River, WA, game-theoretic agent-based behavioral rules were integrated with a Coupled Eulerian-Lagrangian (CEL) Hybrid Model and a computational fluid dynamics (CFD) model to elicit virtual 3-D movement behavior dynamics of outmigrating juvenile steelhead. Hydraulic patterns inherent in natural, free-flowing river reaches (wall-bounded flow) and at dams (free-shear flow), along with hydrostatic pressure, were used as agents to elicit behavior. The resulting simulation model, the Numerical Fish Surrogate (NFS), captures several distinct behaviors observed in 3-D field telemetry data collected in the forebay of Lower Granite Dam. In addition, the NFS captures the percentages with which juvenile steelhead use each of the available passage options around Lower Granite Dam. Agent-based behavior rules were formulated so as to be consistent with existing knowledge of fish mechanosensory systems and existing field data. The NFS, while using many existing concepts and mathematical formulations, is original code and can be easily extended to other aquatic species and ecosystems for which multi-dimensional physicochemical data exists.

1) Civil & Environmental Engineering, Cornell University; Environmental Laboratory (CEERD-EP-W), US Army Engineer Research & Development Center, Columbia River Basin Facility, 458 Evergreen Drive, PO Box 40, North Bonneville, WA 98639; PH (503) 808-4872; FAX (503) 808-4875; e-mail: rag12@cornell.edu

2) School of Aquatic and Fishery Sciences, University of Washington, Columbia Basin Research, 1325 4th Ave., Suite 1820, Seattle, WA 98101; PH (206) 543-4772; FAX: (206) 616-7452; e-mail: jim@cbr.washington.edu

3) Environmental Laboratory (CEERD-EP-W), US Army Engineer Research & Development Center, 3909 Halls Ferry Road, Vicksburg, MS 39180-6199; PH (601) 634-3870; FAX (601) 634-3129; e-mail: nestlej@wes.army.mil

Relational Agent Models — A Framework for Modular and Topology-Independent Simulations

Laszlo Gulyas — gulyas@sztaki.hu
Computer and Automation Institute
Hungarian Academy of Sciences

An increasing number of studies suggest that the results obtained by computational models are dependent on the interaction topology used within them. That is, in case of agent-based models, the outcome could be much different depending on the number and identity of the agents' partners. In the presence of this observation, the correct experimental methodology would be to treat topology as a parameter. That is, to perform tests with as many relevant topologies as possible, and to assess the results' sensitivity towards them, prior to publication.

Nonetheless, most of today's models operate on a fixed, and often arbitrary topology. This is, in part, due to the ancestry of agent-based simulation. One of the most prominent forerunners of the methodology was cellular automaton. Hence the two canonical ABM topologies: the grid and the torus. While at first these seem as appropriate representations of space, it is obvious that they introduce artificial limitations (e.g., on the number of immediate neighbors, etc.). Furthermore, these topologies are often used to implement abstract concepts of proximity, where their suitability is less obvious.

Recent interest in social networks directed attention towards more general interaction topologies, such as random graphs, "small world" networks, etc. Still, even reports on network-based models rarely account for the sensitivity towards the network's properties. Ultimately, this is due to the canonical way (c.f. "design patterns") agent-based models are built. According to this, the topology is referenced throughout the model, affecting both the agents and the core (model/swarm) class. Therefore, experimenting with different topologies would, in practice, mean to re-implement the simulation a number of times, with all the hassles and risks of such an endeavor.

In this presentation two solutions are offered to this problem. First, RePast's space library is outlined, demonstrating its support for "swappable" spaces. While this approach works well for two-dimensional, grid-like topologies, it cannot properly handle generalizations to higher dimensions, nor can it bridge the gap between grid-like and network-based topologies.

The second solution overcomes these latter problems, too, by defining the interaction topology in terms of relations between agents. (For example, an agent can acquire a list of all agents that are in the "Neighbor" relation with it.) Naturally, relations can change over time, thus making dynamic topologies possible. Moreover, in this framework, relations are defined and updated by so-called "contexts". The intent is that contexts encapsulate the handling of interdependent relations, making the approach modular and allowing for easy and flexible experimentation with interaction topologies.

Exploring the Macroscopic Properties of Complex Systems

Paul Harten, Ph.D. — Harten.Paul@epa.gov
Easwari Namboodiri
Sustainable Technology Division, Industrial Multimedia Branch
National Risk Management Research Laboratory
Office of Research and Development
U.S. Environmental Protection Agency
26 West Martin Luther King Drive
Cincinnati, OH 45268

Studying the emergent macroscopic properties of complex systems can be a very complicated task requiring an enormous number of calculations. For some complex systems, the number of calculations necessary for a study may be prohibitive to run on a single PC. By design, these calculations can be distributed across a multitude of PC's and performed concurrently. The cluster of PC's may be constructed, using existing communication standards, over-the-counter hardware, and freely available cluster software.

The complex system studied is an ecological food web. Simulations of the food web are carried-out using the multi-agent rule-based modeling package SWARM developed at the Sante Fe Institute (SFI). In this model, there exist many agents for each of the distinct species of an ecological food web. An agent's behavior is characterized by parameters that determine the rules of interaction of its species. Changing any of the parameters of a species may greatly alter the behavior of this complex system. Thus, a vector made from parameters for all of the species determines the overall behavior of the total system.

The emergent macroscopic properties of these complex environmental systems are explored using the parallel genetic algorithm library PGApack developed at Argonne National Laboratory (ANL). The vector of parameters that determine the behavior of the total system may be viewed as the genome of the system. Then, vectors that optimize specific macroscopic properties in complex systems may be found by natural selection. This same software package PGApack can be used in many different areas of complex systems research.

For research purposes, a 12-node cluster of PC's with MPI (Message Passing Interface) architecture is constructed using the OSCAR software package developed at Oak Ridge National Laboratory (ORNL). This software package is installed on top of the Redhat Linux operating system.

Multi-Agent Virtual Histories of the Development of Differentiated Economic Strategies

Darold Higa — dhiga@cyberianzone.com
University of Southern California School of International Relations

Agent communications, agent cognition and agent strategy selection play a vital role in the evolution of complex behaviors in multi-agent modeling systems. Complex cognitive agents allow models to develop highly differentiated results and show a wider range of outcomes than models that use overly simplistic agents. The ability for agents to create and communicate solutions are important in understanding how stability and instability can develop in the management of complex interactive problems such as resource allocation and usage. Ultimately, these complex cognitive agents and the virtual histories they create may reveal under what conditions successful resource management strategies emerge and proliferate.

Using a modified version of Holland's Learning Classifier System (LCFS), the smallworld simulation records path dependent histories, called virtual histories, of agents interacting with each other and their local environment. The introduction of agents with increased cognitive ability, communication and memory merge the work of Holland, Epstein and Axtell and Alker and Bennett to reveal how the formation and transmission of ideas can mimic patterns found in the real world.

RePast 2.0: Major Changes and New Features

Nicholson Collier and **Thomas R. Howe** — thowe@src.uchicago.edu
Social Science Research Computing
University of Chicago

The University of Chicago's Social Science Research Computing's RePast is a software framework for creating agent-based simulations using the Java language. It provides a library of objects for creating, running, displaying, and collecting data from an agent-based simulation. In addition, RePast includes several varieties of charts for visualizing data (e.g. histograms and sequence graphs) and can take snapshots of running simulations and create QuickTime movies of such. This paper describes some of the major changes and new features provided by the recently released RePast 2.0.

Serialization and Stability Analysis in Swarm Simulations

Paul E. Johnson — pauljohn@ku.edu
Department of Political Science
1541 Lilac Lane, Rm 504
University of Kansas
Lawrence, KS 66044-3177

In the context of a simulation model of networks, this project demonstrates how Swarm's serialization implementation can be done and used. Serialization allows one to save the state of a model—all objects, environments, parameters, etc, and then repeatedly re-start that model. The simulation can be restarted and subjected to various random shocks. By running a model many times, and then subjecting each run to a variety of disturbances, one can obtain a number of interesting insights.

Running C++ Models Under the Swarm Environment

Richard Leow[2] and Russell K. Standish [1 2] — richardl@hpc.unsw.edu.au, R.Standish@unsw.edu.au
1) School of Mathematics
2) High Performance Computing Support Unit,
University of New South Wales,
Sydney, NSW 2052, Australia
http://www.hpc.unsw.edu.au

Objective-C is still the language of choice if users want to run their simulation efficiently under the Swarm environment since the Swarm environment itself was written in Objective-C. The language is a fast, object-oriented and easy to learn. However, the language is less well known than, less expressive than, and lacks support for many important features of C++ (eg. OpenMP for high performance computing application). In this paper, we present a methodology and software tools that we have developed for auto generating an Objective-C object template (and all the necessary interfacing functions) from a given C++ model, utilising the Classdesc's object description technology, so that the C++ model can both be run and accessed under the Objective-C and C++ environments. We also present a methodology for modifying an existing Swarm application to make part of the model (eg. the heatbug's step method) run under the C++ environment.

Agent Based Modeling of Human and Natural Systems and their Interactions

Armin R. Mikler, Michael Monticino, Baird Callicott, **Saqib Khalil** — mikler@cs.unt.edu, monticno@unt.edu, callicott@unt.edu, khalil@cs.unt.edu
Institute of Applied Sciences
University of North Texas
Denton, TX 76203 USA

This paper provides an overview of the simulation of interacting models that combine forest and landscape dynamics with human values and decision-making. It also explores mathematically complex interaction and feedback relationships of integrated models. We attempt to generalize results by application of models to several hierarchical scales across scales and cultures.

Biocomplexity is a new area of multi-and interdisciplinary scientific research focusing on the interactions of various kinds of dynamic living systems. The US National Science Foundation (NSF) has recently designated biocomplexity research as new funding priority. One salient form of biocomplexity is the dynamic interactions between biotic communities, their associated ecosystems and their human inhabitants, which NSF categorizes as Coupled Natural and Human Systems. The authors are members of a multidisciplinary team of researchers, funded by NSF, based at the University of North Texas in collaboration with colleagues at Yale University and Rice University in the United States and the Universidad de Los Andes and Universidad Experimental de Guayana in Venezuela. We are studying the dynamic interaction between vegetation cover, ecosystem function, and human behavior in two sites in Texas and two in Venezuela.

Keywords – Equation Based Modeling, Agent Based Modeling, Swarm, Biocomplexity

H-MAS: A Heterogeneous, Mobile, Ad-hoc Sensor-Network Simulation Environment

Bren C. Mochocki and Gregory R. Madey — bmochock@nd.edu, gmadey@cse.nd.edu
University of Notre Dame
Notre Dame, IN 46703

Ad-hoc Sensor Networks are a feasible and rapidly deployable solution to many common environment monitoring applications. In traditional sensor network design, each unit is capable sensing, processing and storing its own local data, as well as using a multi-hop forwarding scheme to send this data to off-network storage. Thus, each network element must fill at least four different rolls: sensing, processing, communication, and storage, each of which may be further subdivided. As nano-scale miniaturization becomes feasible, including complete homogeneous functionality at each node will become an increasingly difficult problem. Pico-radio devices provide a platform by which these responsibilities can be offloaded to different nodes, but still remain accessible through wireless communication. We have developed a conceptual pico-radio sensor network system, and a corresponding design simulation and evaluation environment, H-MAS, that separates the tasks of processing and storage from the sensor nodes. Through agent-based computer simulation using the SWARM toolkit, we will compare the effectiveness of current medium access, routing, organization, and energy conservation techniques on H-MAS, and derive several heuristics and design rules for various network configurations. The visualization side of H-MAS also provides a convenient way to present the design of MAS systems to non-technical personnel.

Penelope Project: Web-based Textile Production Planning

Matteo Morini — m.morini@labor-torino.it
University of Turin

Penelope is an EU-funded research project based on the textile-oriented production planner SMS (Spinning Mill Simulator), in which partners from different countries across Europe are involved.

The original, prototypical model has been extended in order to deal with weaving mills and incorporated into a framework which is expected, once completed and put into place, to offer a production planning service via a web portal to textile industries.

The optimizing automated planning engine, which is the core of Penelope, consists of an ‘Enterprise Simulator’, a Swarm model of a firm’s supply chain, and a genetic algorithm searching the solutions space. The ES is expected to be generic enough to be able to deal with differently sized and organized firms, both spinning and weaving mills, with a different layout and number of production units, different products-units matching constraints, different planning philosophies, priorities and goals. The search algorithm is Gigi Ferraris’ latest and improved GA implementation, christened Golem by the author.

Being the evaluation of each single strategy the computationally heaviest task being taken over by the system, a middle layer has been inserted between the GA and the ES: the name is Genoma Bucket, and the principle it’s based on is that keeping already-seen genomas-fitnesses pairs stored in a buffer and promptly returned to the GA can significantly save CPU cycles and speed up the search process, achieving reasonable convergence rates in less time.

Work is also being done with the aim of exploiting the intrinsically parallel nature of the search process, during which many different production plans (as GA-fed binary strings) are evaluated by the ES, independently from each other. The most efficient way to distribute this workload on many CPUs (even on different pc's) is being investigated.

Why the Navy Needs TSUNAMI

Michael J. North, Argonne National Laboratory, Argonne, IL USA — north@anl.gov
Michael Rimmer, Navy Warfare Development Command, Newport, RI USA
Charles M. Macal, Argonne National Laboratory, Argonne, IL USA

Admiral Jay L. Johnson, Chief of Naval Operations, U.S. Navy, in his address at the U.S. Naval Institute Annapolis Seminar and 123rd Annual Meeting, in Annapolis, Maryland, on April 23, 1997, said the military is in the midst of "a fundamental shift from what we call platform-centric warfare to something we call network-centric warfare." The Tactical Sensor and Ubiquitous Network Agent-Modeling Initiative (TSUNAMI) directly supports this shift. The Navy Warfare Development Command (NWDC) and Argonne National Laboratory's Center for Complex Adaptive Systems Simulation (CCASS) are collaborating to develop TSUNAMI to support several key NWDC initiatives related to network-centric warfare; expeditionary maneuver warfare; operational maneuver from the sea; and ship to objective maneuver. The goal is to exploit the potential benefits of initiatives such as the Expeditionary Sensor Grid and the capabilities of programs of record such as the Joint Strike Fighter and the Amphibious Assault Vehicle. The NWDC and CCASS are applying agent-based modeling and simulation computing advances to allow TSUNAMI to simulate battle space motion and interaction over real terrain maps; clone sensors and sensor fields; apply rule sets to simulate message traffic; simulate quality of service protocols; and simulate exercise data management choices on a case by case basis to model realistic traffic loadings and network overheads. TSUNAMI agents model interacting blue, red, and neutral forces with complex behaviors and a variety of attributes including varying communications equipment; varying sensing capabilities; varying mobility; varying memory; and varying fuel and battery lifetimes. These agents and their simulated environment will be described in detail; and example model results will be presented.

VSB and SimBuilder - An Analysis of Visual Agent Based Tools

Alessandro Perrone — alex@unive.it
University of Venice

In some sciences, especially in the study of complex systems complex-systems, computer programs play an important role as scientific equipment. In the case of computer simulations, the programs under use, can be seen as experimental devices built in software.
We don't, however, forget that Social scientists are not computer scientists, but their skills in the field will have to improve to cope with the growing subject of social simulation and agent based modeling techniques.

Tools like Swarm, Ascape, RePast can be viewed as an aid to researchers to create a standard tool to build simulations.

But there's a problem. Programming with these tools are not easy, the scientist have to spend a few hours to learn the philosophy of the packages, how to use the libraries, how to call the graphic routines, how to call data to plot, etc. etc, and how to insert them in a model. to avoid this problems, nowadays, there have been created some tools using java language, so the the learning time may be decreased a bit, because it is very easy to find a java programmer, but the problem of learning the use and the philosophy of these tools still remains.

In these years, to help programmers to write their programs, some software houses have programmed packages named Visual tools, in which, with simple clicks of mouse and very short help of the user, anyone can create the graphical Interface of the programs, and, then, at the final stage, the Visual program generate the relative code. Even if the code generate is not "the best code", the output is good. (We do not have to forget that in these year the speed of computers is increased, so a program of several megabytes runs fast). The graphical user interface makes the knowledge of particular programming skills unnecessary.

The VSB and SimBuilder are two of the efforts to ease the development of agent based simulation for scientist. Vsb permits to build simulations in Swarm simulation toolkit, while SimBuilder do the same for RePast environment.

With these tools, building a simulation is then done by adding components from the component palette to the property pane, customizing these components by editing their properties, compiling the project and then running the resulting simulation.

This paper gives an overview of both programs and the environment.

Complex Dynamics of Simple Stock Market Behaviour

Alessandro Perrone — alex@unive.it
University of Venice

Taking inspiration from the seminal work of Day Huang "Bull, Bears and market sheep" (1990), we present a stylized agent-based model of a stock market. Fundamentalist and chartist agent are present and interact in the market where a specialist meets offer and demand schedules adjusting the price to reduce orders imbalance. Heterogeneous valuation of the fair price of the risky asset and simple adaptive behaviour can generate diverse price dynamics, as periodic and quasi-periodic cycles, chaotic time series and persistent deviations from the 'consensus' price. Analysis of the synthetic data generated by the model reveals some finer structure and confirm in a variety of frameworks that non-trivial dynamics is due solely to the deterministic behaviour of agents in the absence of any noisy exogenous shock.

Getting Started with an Agent-based Model

Steve Railsback
Lang, Railsback & Assoc. & Humboldt State University
Arcata, CA
LRA@Northcoast.com; http://math.humboldt.edu/~simsys/

Volker Grimm
Environmental Research Institute Leipzig-Halle
Leipzig, Germany
http://www.oesa.ufz.de/volker

Agent-based models (ABMs) are innately constrained much less than conventional models. ABMs are not constrained to any particular format (e.g., differential equations, matrices), by the need to solve equations, or even to one level of organization. While this freedom from constraints opens up many possibilities, it also can make it difficult to get an ABM project started. Having fewer innate constraints means that more decisions must be made by the modelers; and if the modelers do not succeed in setting their own appropriate constraints then a project can get out of control.

A key ABM design issue is how much complexity of what types to include. What processes should be represented in the model, and which of these processes should be represented as emerging from lower-level mechanisms? It is tempting to let "realism" be the guide for making these decisions, including any processes that we think are important in the real system that we are trying to model. This is a recipe for disaster, however, because there are no limits to how many realistic processes there are and we can spend the rest of our lives adding to the ABM instead of using it to learn about the system. Instead, the guide to how much complexity to include in an ABM should be: what processes are needed in the model to address a specific problem? Specific objectives of the modeling project, defined at the start, are the essential guide to constraining an ABM's complexity and telling us when enough is enough.

A general approach to organizing an ABM-based project so that it productively solves a specific problem (including: completing a dissertation, publishing a paper) has three big steps. First is to define the specific problem(s) your project addresses. Usually the problem involves explaining how some particular dynamics of the system emerge from the agents. Second is to identify the adaptive traits of the agents that give rise to the system dynamics of interest. What agent decisions or behaviors are believed to produce the system dynamics you are trying to explain? Third is to find good models of these adaptive traits: what agent-level algorithms produce the emergent system dynamics in your ABM? This third step typically requires controlled experiments using the ABM as a test-bed, with alternative models of agent traits as hypotheses to be tested.

The above three steps lead to a modeling cycle: building a tentative model, analyzing and testing its results, revising the model, etc. This cycle requires a currency for comparing versions of a model—a measure for evaluating whether some change has significantly improved the ABM. Again, "realism" is a poor choice. Instead, the currency should be the confidence you have that a model explains the system dynamics of interest — how well the model solves the specific problem it was designed for. "Pattern-oriented" methods for analyzing ABMs are particularly useful for establishing this confidence. The modeling literature provides a number of simple tactics that help get the modeling cycle started quickly and productively.

SOME: An Agent-Based Model of Suburban Sprawl

William Rand [1,5] D. G. Brown [1,2] S. E. Page [1,3] R. Riolo [1] M. Zellner [1] L. E. Fernandez [1,2] — wrand@umich.edu
(1) The Center for the Study of Complex Systems
(2) School of Natural Resources and Environment
(3) Dept. of Political Science; Center for Political Studies
(4) Urban, Technological and Environmental Planning
(5) Electrical Engineering and Computer Science, University of Michigan

SOME is an agent-based model (ABM) of land use change at the rural-urban fringe. Land use and cover change at the urban-rural fringe are implicated in a variety of negative ecosystem impacts, including habitat destruction, fragmentation, loss of biodiversity and watershed degradation. The goal of our project is to examine this change and make descriptive and prescriptive claims regarding how to minimize ecological damage. Our immediate goal is to understand how residential agents make decisions on where to live and which dimensions of that decision making process influence settlement patterns on the rural-urban fringe. A better understanding of this process would help guide the design of policy instruments to control the patterns of urban development to improve ecological performance. We call our project SLUCE (Spatial Land Use Change and Ecological effects) and our agent-based model, SOME (Sluce's Original Model for Exploration). The current model was developed in Swarm using agents with heterogeneous preferences and a landscape with heterogeneous properties. The key agents are residents who make decisions about where to live based upon a hedonic utility calculation over a limited number of choices. The landscape has exogenous characteristics like natural beauty and ecological quality. It also has endogenous characteristics like distance to services and neighborhood density. This creates multiple feedbacks within our model which means that residents and service centers change the landscape upon which future residents make decisions. Our model is not very complicated but we have shown that it displays many characteristics present in real cities.

Network Structures, Public Opinion and the Ising Model

Fabio Rojas — fgrojas@midway.uchicago.edu
Department of Sociology
The University of Chicago

Thomas Howe — thowe@src.uchicago.edu
Social Science Computing Center
The University of Chicago

Abstract: This paper uses computer simulation to derive predictions about the effects of network structure on public opinion from a quasi-Ising model of interpersonal influence. Ising models assume that each person sits on the vertex of a rectangular, cubical or otherwise periodic graph. Each actor exerts influence on its neighbors, and there might be a force uniformly exerted on the entire structure. We relax the assumption that interpersonal networks are rectangular, hexagonal or otherwise periodic; patterns of social relations may have a tree structure, a small worlds structure or other network topology. Our paper defines a "quasi-Ising" model of personal influence, discusses the simulation of the model and the relationship between network structures and the degree of agreement in a population with specific patterns of mutual influence.

Complex Motives In Agent Simulation: Proximity Attractors and Their Design

David L. Sallach — sallach@uchicago.edu
Social Science Research Computing
University of Chicago

Within the growing agent simulation community, there are disagreements as to whether agents necessarily must remain simple, so that the dynamics of agent simulation can be fully comprehended, or whether they should be complex in order to represent the subtlety and dynamism of social processes. It is likely that only years of experimentation and analysis will fully resolve the issue, however, at least part of the answer may arise from the art and science of design. The approach of this paper is to: 1) reinforce the necessity of developing models that reflect the complexity of human intentionality, and 2) introduce a dynamic mechanism, an artificial attractor, that integrates multiple domains and constraints. The promise of this approach is its ability to represent complex agent motives using simple mechanisms. KEYWORDS: agent simulation, design, attractor, constraint, artificial social process, motivation, utility

On the Role of Affect in the Evolution of Social Control in Multi-Agent Societies

Matthias Scheutz — mschuetz@cse.nd.edu
Computer Science & Engineering
University of Notre Dame

We present a research strategy to study the effects of various components in an agent architecture (e.g., feedback circuits that can implement various control states) on the dynamics of species interactions in a simulated environment. In particular, we discuss the consequences of behaviors elicited by affective states (like “hunger” or “fear”) for whole agent societies. Construing affective phenomena as processes implemented in part by the agent, but partly also resulting from the agent-environment interaction (including interactions with other agents), we report simulation results that suggest that certain affective states are very likely to evolve in competitive multi-species environments because they lead to behavior that is beneficial at the society level. From this we conclude that further studies along the lines we suggest might reveal evolutionary trajectories from simple individual reflex-like agents to much more complex multi-agent societies, possibly also hinting at why relatively few species have developed a complex deliberative control system.

Novel Technique for PID Tuning by Particle Swarm Optimization

S. Easter Selvan, Sethu Subramanian, S. Theban Solomon — easterselvan@rediffmail.com
Electronics and Communication Department,
Karunya Institute of Technology,
Tamil Nadu,
INDIA

Abstract — An attempt has been made by incorporating some special features in the conventional particle swarm optimization (PSO) technique and its usefulness was tested in a common control application involving PID controller tuning. This algorithm differs from the existing PSO methods in the inclusion of unbiased search approach of the swarm particles without establishing inter-particle communication in the initial phase itself, built-in acceleration mechanism of the particles trapped in the local minima of the objective function, preserving the initial population without reproducing those with favourable costs by cluster unification and finer search phase to identify the global minimum. The results of the aforementioned application in comparison with practical methods are quite encouraging. Index terms — population, swarm particle, local minimum, cost, Euclidean distance.

The JAS (Java Agent-based Simulation) Library

Michele Sonnessa — sonnessa@di.unito.it
University of Turin

The most unpleasant limit of agent-based modeling is the requirement of good skills in computer programming, particularly because it is used by social scientists. Every attempt to create a visual development tool for ABM has shown only one certainty: if you want to realize a valid model, you have to type code. In fact, the agent based models are algorithm intensive applications. Starting from this assumption we tried to create a brand new ABM toolkit, with the aim to hide technicalities to the modeller as much as possible and to create simple and easy-to-use interfaces for the most common tools needed by an ABM simulator. JAS (http://jaslibrary.sourceforge.net) is a java open source project, hosted by the SourceForge directory. It is based on a general-purpose discrete-event simulation engine and borrows many features from other open source libraries. The statistical package and the random number generators are based on the well-known COLT library from CERN; the plotting package is widely founded on the PtPlot tool from Berkeley University, the network interface makes use of the SVG and the XMLRPC java implementations by the Apache Software Foundation. The philosophy in JAS development was to use reliable and powerful open source packages, reducing and homogenizing their complex interfaces in a coherent environment. It has been written in pure Java code and its architecture is largely based on the Swarm paradigm, such that javaSwarm applications can be easily ported to the JAS platform. Among JAS’ features stand out a package for network simulation design (Sim2Web), Gigi Ferraris’ implementation of Genetic Algorithms and Classifier Systems, a forthcoming Neural Networks library, a full support to the XML format.

EcoLab: Agent Based Modeling for C++ Programmers

Russell Standish and **Richard Loew** — R.Standish@unsw.edu.au, richardl@hpc.unsw.edu.au
High Performance Computing Support Unit
University of New South Wales, Sydney 2052, Australia
http://parallel.hpc.unsw.edu.au/ecolab

EcoLab is an agent based modeling system for C++ programmers, strongly influenced by the design of Swarm. This paper is just a brief outline of EcoLab's features, more details can be found in other published articles, documentation and source code from the EcoLab website.

This is not the time or place to debate the merits of C++ over any other object oriented language. If you have chosen C++ as an implementation language for your models because of performance, expressibility, familiarity or compatability with other software libraries, then ABM environments such as Swarm or Repast offer little support to you. In this case, you should consider EcoLab. Scripting EcoLab uses the Classdesc(2) object descriptor technology. This provides a form of object reflection, or the ability to query an object's internal structure at runtime. This may seem obvious to a Java or Objective C programmer, as object reflection is built into the language. How is Classdesc used in EcoLab? The user decodes their entire model as a class. Usually, there will only be one instantiated object of that class (the model). Most model instance variables, and model methods adhering to particular calling conventions are exposed to a TCL interpreter.

Modeling a Swarm of Search and Rescue Robots using Star Logo

Daniel P. Stormont — stormont@cc.usu.edu
Utah State University
Department of Electrical and Computer Engineering
4120 Old Main Hill
Logan, UT 84322-4120

A common problem in a disaster situation, whether natural or man-made, is to quickly identify the locations of potential victims and the locations of hazards that may pose a risk to human rescuers, often without sufficient manpower, during the most critical early hours of a crisis. One way to address this need is to use robots to supplement and assist the first responders to allow them to focus their efforts on rescuing as many victims as possible while minimizing risks to themselves. To be successful, the robots need to be able to communicate their findings back to a command post, cover the maximum area in the minimum possible time, and not hinder or endanger rescuers or victims. The environment is also difficult for the robots to operate in from a standpoint of mobility and communication. Rubble, structural damage, smoke, dust, and fire can interfere with sensors and communications, as well as making it difficult for a smaller robot to move. However, if larger robots are used, they can pose a risk to rescuers and victims through collision or causing structural damage. The best approach would seem to be using a large swarm of small, inexpensive robots that can spread out through the rescue area identifying victims and hazards while maintaining contact with each other to facilitate message passing back to the command post. Behavioral robots that balance between a desire to flock together and a desire to stay away from neighbors can provide the desired area coverage while maintaining communications.

This paper describes modeling work done using StarLogo for a swarm of behavior-programmed robots operating in a simulated rescue environment based on the standard urban search and rescue test course developed by the National Institute for Standards and Technology (NIST). This course is the course that has been used for rescue robot competitions by both RoboCup and the American Association for Artificial Intelligence (AAAI). The paper also describes how the simulated robots could be implemented as physical robots and future plans to build a swarm of small robots to implement this model for the RoboCup and AAAI rescue robot competitions.

Decision Making and Enterprise Simulation With jES and Swarm

Pietro Terna — pietro.terna@unito.it
Dipartimento di Scienze economiche e finanziarie
Università di Torino, Italia

jES, java Enterprise Simulator (formerly jVE, java Virtual Enterprise), is a large Swarm-based package1 aimed at building simulation models both of actual enterprises and of virtual ones. On the first side, the simulation of actual enterprises, i.e. the creation of computational models of those realities, is useful for the understanding of their behavior, mainly in order to optimize the related decisional processes. On the other side, through virtual enterprises we can investigate how firms originate and how they interact in social networks of production units and structures, also in "would be" situations. In both cases, following Gibbons, we have to overcome the basic economic model of the firm, i.e. a black box with labor and physical inputs in one end and output on the other, operating under the hypothesis of minimum cost and maximum profit. Simulation models — such as jES — represent the most appropriate tool to be used in this direction. Agents, in jES, are objects like the orders to be produced and the production units able to deal with the orders. In the same context, there are also agents representing the decision nodes, where rules and algorithms (like GA or CS), or avatars2 of actual people, act. Avatars' decisions are taken asking actual people what to do: in this way we can simulate the effects of actual choices; we can also use the simulator as a training tool and, simultaneously, as a way to run economic experiments to understand how people behave and decide in organizations. This is the big Simon's question. Finally, some recent improvements of jES are outlined in the presentation.

Poster Abstracts

The Effect of Panic on Crowd Dynamics

Gabriel Brown — gbrown2@nd.edu
University of Notre Dame

We read the headlines and shake our heads in disbelief. We see the images on the eleven o'clock news and cannot imagine how it happens. The aggregate damage caused by panicked individual action is many times astounding. In its most benign form, people leaving a venue in a hurry results in a sub-optimal amount of time it takes to evacuate everyone. However at its worst, the collective behavior of panicked people can have fatal consequences. In light of the recent tragedies at nightclubs in Chicago and Rhode Island, we have studied the conditions under which singular actions and panic can create jamming in emergency situations. The study includes how aggregate behavior varies with changes in individual agents' parameters. The simulation is implemented using Swarm so as to control the characteristics of the agents, the environment in which they reside, and the statistical feedback given in varying trials. The agents may vary from how panicked or how calm each person is, or what combinations of different type of people, will vary the outcome. Also investigated is how the density of people in a given space influences results. We notice how a group of calm agents leaving a room in an orderly fashion exit more quickly as compared to when a crowd rushes the door. We extend the notion of an emergency situation by simulating smoke in a room and correspondingly give agents a limited view of exits. We also add to the simulation problems akin to those reported in Chicago in which people ran to the door they entered, leaving other viable exit doors underused. Additionally, we alter the structure from which these agents are attempting to escape. We show how different room sizes, shapes and configurations influence results. Even though each person's actions cannot be altered (their panic "unlearned") number, types, and sizes of exit doors can be changed to prevent future tragic consequences.

Discovering Algorithms that Generate Hamiltonian Graphs

Scott Christley — schristley@mac.com
Portland, OR, USA

Cycles are an important conceptual property in multi-agent systems. Cycles appear quite frequently in agent modeling as agents are often coupled together such that they interact with other agents and the environment around them. Agent systems can be represented by graphs and graphs that evolve over time; and the existence of cycles in these graphs define a feedback recurrent architecture which is an enabling feature for multi-agent systems to exhibit complex and emergent behaviour. Hamilton cycles are an important subset of cycles in that they exhibit complex structure plus they are fundamental to the mathematical concept of isomorphism. Hamiltonian graphs have the property of being somewhat easy to generate but hard to verify; that is, discovering a Hamilton cycle within a given graph is known to be NP-complete while generating a simple graph guaranteed to contain a Hamilton cycle can be accomplished in linear time. Yet production of Hamiltonian graphs is just the inverse of the search problem; that is, producing the full set of Hamilton cycles is still intractable because there are a factorial number of them. Genetic Programming evolves a population of programs with genetic operators according to a specified fitness function. The well-known form is

for the individual agents to be represented by acyclic trees whereby subtrees are exchanged during crossover; however, generating cycle graph programs will require that the issues of iteration and recursion be handled because they are the programmatic concept of cycles. This article investigates whether a Genetic Programming system can discover algorithms that generate Hamiltonian graphs, how would such a system be defined, what are its properties, and what are the limitations of such a system. It is conjectured that the power set of the subclasses of all Hamiltonian graphs may be related to grammars that define automata or structures for programs.

Topology and Evolution of the Open Source Software Community

Yongqin Gao — ygao1@nd.edu
Department of Computer Science and Engineering
University of Notre Dame
Notre Dame, IN 46556

The Open Source Software (OSS) development movement is a classic example of a social network; it is also a prototype of a complex evolving network. By continuously collecting developers and projects information from SourceForge for two years, we have sufficient data to infer the dynamic and the structural mechanisms that govern the evolution and topology of this complex system. We use this information in three steps. First, we analyze the empirical data we get from SourceForge to get statistics and topological information of the OSS developer social network. Also we extract the evolution of parameters by inspecting the network by time. This gives us knowledge about what the complex network looks like and data about the characteristics of the system. Second, we generate a data model to model the evolution of this network. Finally we simulate the evolution of the OSS developer complex network using Java/Swarm and tune the simulation to match the real data. In simulation, we used several different model to find the best fit model for real data. These models include random network, scale free network and scale free network with fitness. Through simulation we can also verify our data model for this network and get predictions of the evolution of the complex network.

How Much Is Too Much? — What Programming Skills Are Really Needed to do ABM?

Laszlo Gulyas — gulya@latte.harvard.edu
Computer and Automation Institute
Hungarian Academy of Sciences

Sandor Bartha
Lorand Eotvos University
Budapest, Hungary

Recent discussions in the agent-based modeling community have showed disparate opinions about the level of programming skills to be expected from modelers. Everybody seems to agree that the more skill candidates have, the better. However, most of today's students lack these capabilities and developing them requires substantial efforts from the adventurous entrepreneur. Therefore, lowering the requirements would help agent-based modeling becoming a more widely accepted methodology.

Some suggest that with the spread of general computer literacy, the problem outlined above will eventually go away, sooner than we would imagine. Others argue that the problem is inherently social: mastering mathematics or statistics is not in the least easier than learning to program, still no aspiring scientist can afford to avoid it.

It would be hard to deny any of these arguments. Nonetheless, we argue that today's requirements can be lowered. This statement is not very surprising either. Various model building tools (such as NetLogo or AgentSheet) demonstrate that by limiting the 'space' of possible models, the task of modeling can be efficiently assisted. The real challenge is to bridge the gap between the potential open-ended nature of Swarm-alike modeling environments (e.g., Swarm, RePast, Ascape) and the ease of use provided by the former frameworks.

Graphical model building interfaces for general ABM platforms, such as SimBuilder (for RePast) and the Visual Swarm Builder (VSB) are attempts to achieve exactly this. Still, they impose certain limitations on the modeler and require a certain level of programming.

Mass-action vs. the Gauntlet: Survival of Migrating Prey in Predator Fields

Eliezer Gurarie — elieza@u.washington.edu
Interdisciplinary Program in Quantitative Ecology and Resource Management
University of Washington
Seattle, WA 98195

Ecological theory traditionally describes predator-prey interactions in terms of a law of mass action in which the prey mortality rate depends on the density of predators and prey. In such models, the mortality rate is a function of the exposure time of the prey to predators. However, observations on migrating prey through a field of predators (eg. ocean-bound juvenile salmon among piscivors) reveals mortality depends

mostly on distance traveled and only weakly on travel time. This is due to the effects of explicit spatial structure. If, for example, prey migrate through a stationary gauntlet of predators, mortality depends strongly on distance. At the opposite extreme of random prey and predator motions, mortality is time-dependent. These effects have been explicitly described by an analytical mean-free path model based on predator-prey encounter areas and relative velocities. I use a Swarm simulation to test the analytical model and to further explore the time and space dependencies of survival with heterogeneous agent structure and behaviors.

Toward an Agent Based Model of Open Source Software Development

Chris Hoffman — choffma1@nd.edu
University of Notre Dame

Open Source Software (OSS) development maintains the interest of researchers worldwide. A number of teams have begun investigating OSS as a self-organizing, collaborative social network. Of particular concern are its implications for proprietary software development firms who may adopt open source practices should they prove superior to centrally managed corporate structures. Also thought provoking is the possibility of extending discoveries of the underlying mechanisms of OSS development to explain the mechanics of social networks in general. Decision or communication structures in OSS forums may correlate to those in such varied social networks as the Internet Movie Database or even terrorist cells.

As the basis for further research of OSS and other social networks, we have coded an agent-based simulation using Swarm and Oracle to model the development of the OSS network. We discuss here the design of our simulation, as well as recommendations for future improvement.

Swarm-based Modeling of Speciation

Todd Hoffenberg — thoffenb@nd.edu
University of Notre Dame
Notre Dame, IN 46556

Speciation, to evolutionary biologists, is a well-observed phenomenon. However, because of the long time scales involved in this process and relative complexities of real evolutionary processes, it has been very difficult to induce. Modeling such a complex system requires a distributed approach. With a simple complex-adaptive-system where little besides a reproductive capacity, lifespan, and inherent locality are the only parameters, eventual behaviors to optimize reproduction will emerge. Keeping track of species divergence is possible through a taxonomical ancestry tree, a useful tool for several Swarm-based simulations. With a successful distributed architecture, evolutionary scientists could begin to paramaterize speciation and co-adaptation. The initial implementation, similar to anthill colony simulations, uses agents capable of releasing and detecting pheremone gradients with rule-based chemical counteraction. The distributed environment is achieved through Java RMI.

Web-based Molecular Modeling Using Java/Swarm, J2EE and RDBMS Technologies

Yingping Huang, Gregory Madey, Xiaorong Xiang and Eric Chanowich — yhuang3@nd.edu, gmadey@nd.edu, xxiang1@nd.edu
Department of Computer Science and Engineering
University of Notre Dame
Notre Dame, IN 46556

Steve Cabaniss — cabaniss@unm.edu
Department of Chemistry
University of New Mexico
Albuquerque, NM 87131

This paper describes the application of Java/Swarm, J2EE and RDBMS technologies to the area of web-based scientific simulation. One such scientific simulation is on the behavior of natural organic matter (NOM). NOM is a heterogeneous mixture of organic molecules found in terrestrial and aquatic environments — from forest soils and streams to coastal rivers and marshes to the open sea. NOM plays a vital role in ecological and biogeochemical processes. In this paper, we present an agent-based stochastic simulation of NOM transformations including biological and non-biological reactions, as well as adsorption and physical transport. It employs recent advances in web-based development environments such as Java 2 Enterprise Edition (J2EE), and scalable web-based relational database management systems (RDBMS), to improve reliability and scalability of the stochastic simulations and to facilitate analysis and data mining of the resulting large datasets. The NOM simulation system may be useful in many areas including chemistry, geology, microbial ecology and environmental science.

Decentralized Message Routing in Mobile Networks using StarLogo

Ryan Kennedy — rkenned1@nd.edu
Michael Kirkpatrick
University of Notre Dame

Decentralized systems have existed in nature for thousands of years. Each time you look at a flock of birds in the sky or at ants foraging for food, you witness a decentralized system. They rely of the concept local agent to agent interaction; there is no global communication. This simple local communication has the capability to expand to many of our existing systems, such as in computer networks. Not only would it be more reliable, but it also has the potential to be more efficient. The future of decentralized systems in our society is promising.

This report describes using StarLogo to simulate some simple decentralized systems. The simulations simulate a message traversing through a system of agents and measure message saturation when the message has expired from the system. Individual agents can only receive and send the message for an user-controllable number of time steps. The results show that for either relatively high densities of agents or for a high number of time steps to retain the message, the message saturation is near 100%. This is exciting information when one considers that parallel uses of these simulations could mimic a network of cell phones or even mimic many tiny robots exploring a distant planet. The applications of using local interactions to perform global tasks are virtually endless. This work will show that simple agents obeying simple rules can lead to grand achievements.

Using Swarm Intelligence to Broadcast Messages in Highly Mobile Ad Hoc Networks

Michael Kirkpatrick and Gregory Madey — mkirkpatrick@acm.org
University of Notre Dame

This paper explores the potential for the efficient broadcasting of messages in highly mobile multi-hop ad hoc networks by treating the nodes in the network as autonomous agents for which only a simple set of rules governing the passing of messages from one node to another is defined.

The general problem presented by highly mobile systems is one of coordinating communication between nodes in settings in which the details of their connections to each other the portions of the network that are within radio range at any given time can change. Nature provides us with examples such as birds flying in formation of mobile, independently operating agents that seemingly work together to perform tasks in a highly efficient manner without complex communication networks and without global knowledge of the locations of individuals. This paper applies this paradigm to the problem of message broadcast in ad hoc networks formed by highly mobile agents. A “biologically inspired” heuristic for broadcasting is introduced. This heuristic has low computational overhead making it especially suitable for “power-limited” systems.

CA Model For Limb Chondrogenesis Based on Reaction-Diffusion and Cell-Matrix Adhesion

Maria Kiskowski — mkiskows@nd.edu
University of Notre Dame

Developmental patterning; the establishment of nonuniform, nonrandom arrangements of different cell types within tissue masses, results from an interplay of changing gene expression and physical changes. Computational models, which can organize and integrate over time vast numbers of molecular interactions and cell behavioral changes, are ideal to represent such complex dynamics.

This poster focuses on a discrete, lattice-based CA model for behavior of limb bud precartilage mesenchymal cells undergoing chondrogenic pattern formation. In our “agent-oriented” CA model, cells are represented by points on a lattice and are assigned simple rules motivated by experimental findings. The rules include random cell motion, production and lateral deposition of a substratum adhesion molecule (SAM), production and release of a diffusible growth factor (“activator”) that stimulates production of the SAM, and another diffusible factor (“inhibitor”) that suppresses the activity of the activator. Parameters are identified for which the system exhibits nodular patterns that resemble those of leg cell cultures, including number, distribution and spacing of cell condensations. This reference system was then studied experimentally, including subjecting it to cell dilution, transient exposure to exogenous activator, suppression of inhibitor, and constitutive activation of SAM production. There was good correspondence between in silico and in vitro experimental results.

Swarms of Swarms — An Extension of the Heatbug Tutorial

James Kramer — jkramer3@nd.edu
University of Notre Dame
Notre Dame, IN 46556

The notion of a "swarm of swarms" (i.e., a hierarchy of swarms) is a strong theoretical concept meant to provide an explicit, measurable, and examinable link between the microlevel constituents in a complex system and the properties and actions exhibited by the macro-level system. Proposed examples range across such diverse topics as the modeling of biological organisms, techniques employed by the U.S. Army in both organization and battle tactics, the design of object-oriented programming languages, historical societal evolution and interaction, and the structure of economic models, to name just a few. The design of the Swarm package incorporates the idea of hierarchical swarms as a major design feature. However, currently there are very few available model implementations at the tutorial level that utilize this capability. This paper examines some possibilities and ideas for implementing multi-level swarms, with the end goal of serving as both a research vehicle and a form of mini-tutorial. With this in mind, extensions of the well-known Heatbug tutorial code are designed and implemented.

Constructing Federations from Simple Agents and Environment Contexts

Dmitri O. Kondratiev, Vasily V. Suvorov — dkondratiev@luxoft.com, vsuvorov@luxoft.com
R&D, Luxoft, 9 -b Dmitrovskoye shosse
Moscow 127434, Russia

Multi-agent approach to simulation modeling suggests an environment where agents can communicate and evolve. Properties of agents and environments vary considerably depending on modeling domain. On the other hand, such properties of agents as emergent behavior and adaptation to environment through evolution are of general value in multiple domains. To realize these fundamental properties the core set of agent and environment features needs to be defined. In this paper we make an attempt to define this core set of agent/environment features and investigate the possibility to realize them with elementary building blocks that agents can be built from. We propose core agent architecture that should allow building agent federations according to different decomposition principles from simple agents (later in this paper referred to simply as agents). Federations of interest for this research include agent swarms and hierarchies of recursive and selfsimilar agents or "holons". As described in holon is "a biological or sociological structure that is stable and coherent and that consists of further holons that function according to similar principles. No natural structure is either "whole" or "part" in an absolute sense, instead every holon is a composition of subordinate parts as well as a part of a larger whole." With well-defined building blocks of software architecture a group of several holonic agents may form a superholon. This super-holon looks to the outside world like a single holon agent defined in terms of the same architectural blocks. On the other hand, core architecture proposed here does not impose any additional constraints on autonomy of subholons that holonic architecture does in terms of goals sharing and resource management between super-holon and sub-holons.

Organizational Decision Chemistry On A Lattice

Alessandro Lomi — alx@economia.unibo.it
University of Bologna
Faculty of Economics

Stefano Cacciaguerra — scacciag@CS.UniBO.IT
University of Bologna
Department of Computer Science

We use NetLogo to design a relatively simple multi-agent environment to explore organizational decision processes. Our work builds on the original insight of the "Garbage Can Model" of organizational choice (GCM) proposed by M. Cohen, J. March, and J. Olsen (1972). According to this model organizations are viewed as crossroads of time-dependent flows of four distinct classes of objects: "problems," "solutions," "participants" and "opportunities." Collisions among the different objects generate events called "decisions." In our NetLogo-based reconstruction of the GCM, the type of decision is determined by the relative levels of energy accumulated by "participants" and "opportunities" up to the moment of collision. No attempt is made to reproduce all the features of the original model while some features of our representation are not present in the original model and are included to cast new light on specific aspects of decision processes in organizations. The model should be considered as a highly preliminary attempt to represent organizational decision processes when agents live in a structured socio-physical space, and are capable of reproducing, i.e. of creating identical copies of themselves. In its current state of development the model serves mainly didactic purposes. For example, the simulator could be used as a computer-based learning environment to introduce students with an interest in institutions to central issues in the study of organizational decision processes, and to discuss the corresponding representation problems that are typically associated with these issues.

Swarm Simulations of the Power Law Distribution Models

Ewa Misiolek — emisiole@cse.nd.edu
Department of Computer Science and Engineering
University of Notre Dame
Notre Dame, IN 46556 USA

Power law distributions have been observed and investigated by researchers for many decades, reaching back over a hundred years to Pareto and his observation of power law distribution of the size of income, followed by power law distribution of word frequencies, city sizes and many other in the areas of biology, chemistry, linguistics, economy and most recently computer science. Since power law distributions play such an important role in many areas it is natural to investigate their properties and construct models leading to their emergence. Many such models have been proposed but no definite conclusions were reached. Most recently, results in computer science demonstrate power law distribution in the number of links between web pages in the World Wide Web network, other research suggests power law for the file size distribution. These observations inspire computer scientists to attempt the construction of power law models followed by their computer simulations. Here we use several Swarm simulations to investigate such models. We base our simulations on existing models where incremental growth and preferential attachment are the key ingredients for the emergence of power laws as well as expand those to include new variables. A new model without the incremental growth requirement is also proposed.

Application of Ant-Colony Optimization to Run-time Systems in High Performance Computing

Arun Rodrigues — arodrig6@nd.edu
University of Notre Dame

Increasing demands in the field of high performance scientific computing have lead to supercomputers of increasing sophistication and complexity. Utilizing a number of techniques to extract parallelism and maximize performance, recent trends in supercomputing have come at the expense of programmability. The knowledge and effort required to perform optimization on a modern supercomputer often detracts from the chief goals of the user. The future generations of supercomputers will have to provide not only high performance, but also high programmability and productivity. One mechanism to achieve this is the use of automatic introspection mechanisms. This work explores how ant-colony type optimizations can be used to dynamically perform low-overhead introspection and optimization in a high performance computing environment.

Simulation Bacterial Conjugation with Swarm

Keefe Roedersheimer — kroeders@nd.edu
Computer Science and Engineering
University of Notre Dame

Bacterial conjugation is a process of horizontal gene transfer in certain types of bacteria. A small ring of DNA, called a plasmid, exists in these bacteria and can express some genetic trait, such as antibiotic resistance. If a donor bacteria (one possessing a plasmid and capable of passing it on) comes into physical contact with a recipient bacteria (a bacteria capable of receiving the plasmid that has not yet done so) it is possible that a link between the two bacteria will be created and the plasmid will duplicated itself and spread to the recipient bacteria. Interestingly, donor and recipient bacteria need not be of the same species for this to occur. Y. Wu (2002), the conjugation rate of bacteria was modeled by fitting established conjugation models to experimental data. This project modeled this process from the bottom up using swarm and the rate constants determined in the mathematical modeling. The resulting simulation populations were compared against the data from the experiments for verification.

Modeling Supply Networks as Complex Adaptive Systems

William J. Sawaya III — wsawaya@csom.umn.edu
Operations and Management Science
University of Minnesota

There has been a great fervor of interest in the relationship between trading partners in business relationships over the last few years. Much of the research has focused on the dyad between organizations or a chain of such linkages. But the nature of such relationships is typically much more complex than a dyad or a chain of dyads; there are generally multiple suppliers and customers of each organization — forming a network of relationships. The idea of complex adaptive systems may be used to address the complexities that arise in such a system. This research contributes to the idea of conceptualizing a supply network as a complex adaptive system by designing a simulated model of supply network. After the initial stages of development, it will allow testing of some of the principles of complex adaptive systems in a supply network environment. This research addresses the specific question of the impact of different policies for sharing information between organizations on a supply network performance. Specifically, organizations are defined as agents with decision schema and access to varying types of information from their trading partners. By using swarm to simulate these relationships, it is possible to build a more realistic representation of the supply network and observe how the interaction of the various organizational agents affects both measures of their own performance as well as that of the network collectively.

Exploring Alternate Topologies in Artificial Neural Networks

Timothy Schoenharl — tschoenh@nd.edu
Department of Computer Science and Engineering
University of Notre Dame
Notre Dame, IN 46556 USA

In this project we explored a novel topology of artificial neural networks, using the techniques of agent-based modeling. Common neural network topologies, such as feed forward networks, do not reflect recent advances in the study of complex biological networks. We have used techniques from agent based modeling to create a neural network that develops a scale-free topology. We modeled the neurons as agents, which enabled us to create and remove connections between the neurons, dynamically adjusting the topology of the network. We built on the work of Barabasi by developing the network structure via local interaction of the neuron agents, utilizing preferential and fitness-based attachment. Recent biological models of the development of the brain suggest significance in both attachment and detachment of neuronal connections, and these results have also been incorporated into our model. We demonstrate our simulation, modeled using the RePast framework.

Modeling Intercellular Communication Patterns

David Tuck — david.tuck@yale.edu
Yale University

Quantitative high throughput molecular biology technologies combined with the rapid ongoing explosion of mammalian genomic and proteomic data have led to growing interest by cancer biologists in simulations to predict the dynamics and mechanisms of biological systems. For example, the most commonly accepted model of tumor progression is clonal evolution, the successive acquisition of mutations leading to a single dominant clone characterizing a tumor. However recent work suggests that genetic and spatial heterogeneity in tumor cell populations are maintained both during early tumor development and throughout tumor progression. In a separate body of work, mathematical models of intercellular communication have been used to assess spatial patterns of cellular behavior based on juxtacrine, paracrine, or autocrine cell-cell signaling, and both negative and positive regulatory feedback loops. Ongoing work, combining published results from these two areas of research, is aimed at modeling the role of intercellular communication in the maintenance of spatial and clonal/genetic heterogeneity in tumor progression. The feasibility of using agent based modeling is being explored in a pilot project using Repast to assess whether this modality will be an appropriate addition to existing tools and provide a foundation for iterations involving modeling and bench research. Preliminary results and approaches are discussed.

Dynamic Fitness in Barabasi-Albert Networks

Michael Ward — mward3@nd.edu
University of Notre Dame

The Barabasi-Albert model of network growth extends earlier random models with the concept of preferential attachment, in which qualities of a node (number of links and fitness) determine the probability that it will receive a link. However, each node's fitness remains static through the life of the graph. Fitness does not seem like something that should remain fixed for the lifetime of the node in question.

We used the Swarm toolkit to experiment with various types of dynamic fitness and observe the effects they had on the scale-free properties of a BA graph. Aging of nodes causing a reduction of fitness has already been studied, and its effects on the graph have been noted. However, the age of a node is no guarantee that it is less fit. For example, in a competitive world such as the Internet, when one node or site gains an advantage, others may try to imitate its success.

This concept of dynamic fitness appears elsewhere. For example, population distributions are in constant flux, with the identities of the most densely populated areas changing over time, while the number of such places remains small. This property seems to carry over to graphs that follow a similar power-law distribution.

Modeling the Spread of Computer Viruses in Scale-free Networks

Gabe Williams — gwillia1@nd.edu
University of Notre Dame

Abstract: Recent studies into the topology of the Internet have shown that it is essentially a scale-free network consisting of a large number of sparsely connected nodes and a small number of highly connected nodes (hubs). This, coupled with existing studies that treat the spread of computer viruses like the spread of epidemiological viruses, will allow us to more accurately model the spread of viruses on the Internet using Swarm (ABM) models. In addition, it will allow us to establish better techniques in defending against network attacks. Each node in these models is treated as an independent agent that has a set of connections to other nodes in the swarm. Most of the nodes are connected with anywhere from 5-10 other nodes. There are a few hubs, however, that are connected to hundreds of other nodes. Along with the connectivity set, each node can be in one of three states: infected, susceptible, or protected. The spread of viruses is dependent on the number of infected and susceptible nodes and the amount of time that it takes for the virus to be spread. The user, to tailor the experiment to many different types of viruses, can specify each of these parameters.

Exploring Performance Improvement For Java-based Scientific Simulations That Use The Swarm Toolkit

Xiaorong Xiang and Gregory Madey — xxiang1@nd.edu, gmadey@nd.edu
Computer Science and Engineering
University of Notre Dame

There is growing interest in using the Java programming language for scientific and engineering applications. The is because Java offers several features, which other traditional languages (C, C++, Fortran) lack, including portability, "garbage collection" mechanism, build-in threads, and the RMI mechanism. However, the historic poor performance of Java stops it from being widely used in scientific applications. Although research and development on Java resulting in JIT compilers, JVM improvement and high performance compilers, have been done much for runtime environment optimization and significantly speeded up the Java programs, we still believe that creating hand-optimized Java source code and optimizing code for speed, reliability, scalability and maintainability are also crucial during program development stage. Natural organic matter (NOM), a mixture of organic molecules with different types of structure and composition, micro-organisms and their environment form a complex system. The NOM simulator is an agent-based stochastic model for simulating the behaviors of molecules over time. The simulator is built using Java programming language and the Swarm agent-based modeling library. We analyze the NOM simulation model from several aspects: runtime optimization, database access, objects usage, parallel and distributed computing. The NOM simulation model possesses most of characteristics which general scientific applications have. These techniques and analysis approaches can be generally used in other scientific applications. We expect that our experiences can help other developers using Java/Swarm to find a suitable way of achieving higher performance for their applications.

A Docking Experiment: Swarm and Repast for Social Network Modeling

Jin Xu — jxu1@cse.nd.edu
University of Notre Dame

Yongqin Gao — ygao1@nd.edu
University of Notre Dame

Axtell, Axelrod, Epstein and Cohen (1996) describe a docking or alignment process and experiment for verifying simulations. By comparing simulations built independently using different simulation tools, the docking or alignment process may discover bugs, misinterpretation of model specification, and inherent differences in toolkit implementations. If the behavior of the multiple simulations are similar, then verification confidence is increased. North and Macal (2002) reported on such experiment using Mathematica, Swarm and RePast to simulation the Beer Distribution Game (originally simulated using system dynamics simulation methods).

In this paper, we present the results of docking a Repast simulation and a Java/Swarm simulation of a social network model of the Free/Open Source Software (F/OSS) community. Data about the SourceForge F/OSS developer site has been collected for over 2 years. Developer membership in projects is used to model the social network of developers. Social networks based on random graphs, preferential attachment, preference attachment with fitness, and aging of nodes are modeled and compared to collected data. The simulations grow "artificial societies" representing the SourceForge developer/project community. As a by-product of the docking experiment, we provide observations on the advantages and disadvantages of the two toolkits for modeling such systems.

SwarmFest 2003 Organizing Committee

James Anderson, jim@cbr.washington.edu, University of Washington

Roger Burkhart, burkhartrogerm@johndeere.com, John Deere

Paul Box, paulbox@cc.usu.edu, Utah State University

Randy Burge, burge@swarm.org, Swarm Development Group

Marcus Daniels, mgd@swarm.org, Sante Fe Institute

Paul Johnson, pauljohn@swarm.org, University of Kansas

Alex Lancaster, alex@swarm.org, University of California, Berkeley

Greg Madey, gmadey@nd.edu, University of Notre Dame

Steve Railsback, LRA@northcoast.com, Humboldt State University

Glenn Ropella, gepr@swarm.org, Swarm Development Group

The Swarm Development Group (SDG) located at www.swarm.org, is a 501c(3) not-for-profit organization dedicated to advancing the state-of-the-art in multi agent based simulation through the continued advancement of the Swarm Simulation System and support of the Swarm user community.

http://www.nd.edu/~swarm03