

# SwarmFest 2002 program schedule

The program schedule is subject to change (*Last updated: 2002/03/27 04:56:10 [UTC/GMT]*). A list of oral presentations appears below.

Friday 29 March 2002	
<b>Opening Reception and Dinner Location: Kane Hall, Walker-Ames Room, University of Washington campus</b>	
<b>6:15 - 7:00 pm</b>	<i>Reception</i>
<b>7:00 - 9:00 pm</b>	<i>Dinner</i>
<b>9:00 - 10:00 pm</b>	<p style="text-align: center;"><i>Key Note Address: "Agent based economic models on the Santa Fe Trail"</i></p> <p><i>Michael W. Brown: Member of the Particle Economics Research Institute and Governor of National Association of Securities Dealers; former Chairman of Nasdaq Stock Market Board of Directors and CFO of Microsoft Corporation.</i></p>

Saturday 30 March 2002	
<b>Location: Mary Gates Hall, Room 241, University of Washington campus</b>	
<b>8:00 am</b>	<b>Continental Breakfast:</b> <i>Pastries, coffee, tea, hot chocolate</i>
<b>8:30 am</b>	<b>Introduction</b>
<b>9:00 am</b>	Steve Railsback (Lang, Railsback & Associates), "More Fishy Business: Using Swarm Models to Test Theories of Fish Ecology"
<b>9:30 am</b>	Paul Johnson (University of Kansas), "Top down and bottom up dynamic scheduling in Swarm: a comparison of implementations."
<b>10:00 am</b>	<b>Break:</b> <i>Coffee, tea, soft drinks</i>
<b>10:15 am</b>	Zoe-Vonna Palmrose (University of Southern California), "Particle Economics Research Institute: the PartEcon Model"
<b>11:00 am</b>	Riccardo Boero (University of Surrey), "gpSwarm: Java Genetic Programming using the Swarm libraries"
<b>11:30 am</b>	Matteo Morini (LABORatorio R. Revelli), "SMS: A Spinning Mill Simulator"
<b>12:00 pm</b>	<b>Lunch</b> <i>[on your own]</i>
<b>1:30 pm</b>	Paul Box (Utah State University), "A Computational Environment for Evaluating the Interaction of Grazing Animals and Landscapes"
<b>2:00 pm</b>	Yuya Sasaki (Utah State University), "Experimental Spatial Economics"
<b>2:30 pm</b>	<b>Break:</b> <i>Cookies, brownies, soft drinks, coffee, tea</i>
<b>3:00 pm</b>	Michael North (Argonne National Laboratory), "The Beer Dock: Three and a Half Implementations of the Beer Distribution Game"
<b>3:30 pm</b>	Raj Divakaran (Utah State University), "Potential Of Ant Colony Optimization To Satellite Image Classification"
<b>4:00 pm</b>	Gianluigi Ferraris (Torino, Italy), "Classifier Systems in Swarm: a general purpose software."
<b>6:00 pm - 12:00 am</b>	<i>Band, Beverages &amp; Banter</i>

Sunday 31 March 2002	
<b>Location: Mary Gates Hall, Room 241, University of Washington campus</b>	
<b>8:00 am</b>	<b>Continental Breakfast:</b> <i>Pastries, coffee, tea, hot chocolate</i>
<b>8:30 am</b>	James Anderson (University of Washington), "A decision algorithm with memory and learning"
<b>9:15 am</b>	Pietro Terna (University of Torino), "Improved Java Virtual Enterprise (jVE) in Swarm"
<b>10:00 am</b>	<b>Break:</b> <i>Coffee, tea, soft drinks</i>
<b>10:15 am</b>	Paolo Patelli (Santa Fe Institute), "Modularity and Robustness in Organizations"

Sunday 31 March 2002

<b>10:45 am</b>	Tom Howe (University of Chicago), "Recent Developments in Repast"
<b>11:15 am</b>	Marcus Daniels (Santa Fe Institute), "IMA: The Integrating Modeling Architecture"
<b>11:45 am</b>	<b>Lunch</b> <i>[on your own]</i>
<b>1:15 pm</b>	Steve Railsback (Lang, Railsback & Associates), "Can Agent-based Modeling Contribute to Scientific Theory?"
<b>2:00 pm</b>	Alessandro Perrone (University of Venice), "VSB 2.0: Visual Swarm Builder"
<b>2:30 pm</b>	<b>Break:</b> <i>Cookies, brownies, soft drinks, coffee, tea</i>
<b>2:45 pm</b>	Greg Madey (University of Notre Dame), "Using Java and Swarm to Build a Stochastic Simulation of Natural Organic Matter (NOM) and Microbe Interactions in the Soil"
<b>3:15 pm</b>	<i>Wrap up discussion</i>
<b>4:00 pm</b>	<i>End of conference</i>

(Note: abstracts are listed alphabetically by presenter, not program schedule order.)

**A decision algorithm with memory and learning**

James J. Anderson <jim@cbr.washington.edu> (Columbia Basin Research, School of Fisheries, University of Washington)

In game theory, a player selects between alternative opportunities to obtain the best outcome. This process requires that the player has knowledge of the opportunities, their values or rewards, the probabilities of obtaining the rewards, and the cost of playing. In early decision theories, players have this knowledge a priori. For example, in marginal foraging theory an animal leaves a patch when the consumption rate declines to the average over all patches. However, in agent-based models, we prefer agents not be endowed with omniscience. Here I explore the properties of a simple model in which an agent learns through encounters or events, imperfectly remembers what it learns, and makes decisions by considering past and recent events. I explore a three-parameter version of the model with a player that continually decides between two states that present different, random patterns of opportunities. Under a wide range of parameters the player first explores both states and then settles into an optimum decision pattern. However, under other ranges of parameters, the player continually switches states, resulting in a suboptimum behavior. I conclude that the model, with its game and learning theory foundations, and its simplicity, may have wide application in agent based models. It is currently being applied to fish foraging (Anderson, 2002) and terrorist behavior (Silverman 2001).

**gpSwarm: Java Genetic Programming using the Swarm libraries**

Riccardo Boero <R.Boero@surrey.ac.uk> (Department of Sociology, School of Human Sciences, University of Surrey, Guildford, United Kingdom)

This paper aims to present a new tool, called 'gpSwarm'. for making Genetic Programming (GP). GP is a computational technique aiming to generate computer programs able to solve problems without being explicitly programmed for them and it belongs to a sub-field of Artificial Intelligence that studies topics related to machine learning. gpSwarm is based on Java and on the Swarm libraries: it uses Swarm for managing the evolution algorithm and it uses Java as the language of possible solutions. gpSwarm is characterised by some advantages in comparison with other tools, mainly the opportunity to use different kinds of variables and the easy management of solutions space. Finally some examples are presented, illustrating how to use the tool and how to integrate it with Agent Based Models.

**A Computational Environment for Evaluating the Interaction of Grazing Animals and Landscapes**

Paul Box <paulbox@cc.usu.edu> (Utah State University) and Juan Villalba (Utah State University)

Ecosystem diversity is largely a result of the complex, non-linear interactions of the constituent members and communities that make up the ecosystem. One example of these interactions is between grazing animals and the plants that they graze on. Herbivores select plants based on real or perceived needs for calories, proteins, vitamins, minerals, or any other part of a well-balanced diet. The cues for herbivore grazing behavior is largely a result of post-ingestive feedback, which in turn is dependent on chemical reactions within the gut based on recent diet. The grazing behavior in turn affects the composition of plant communities in ways that may not be directly predictable from short-term observation, through selective cropping, trampling, seed transport, etc. Plants and grazers adapt over time in ways that either foster or discourage interaction, eliciting counter-adaptations in a cycle that can produce very complex webs of dependencies. Modeling such a system is complicated by the multiple feedbacks, and disparity of temporal and spatial scales of processes. A computational environment that addresses these issues can be constructed using principles of agent-based modeling, cellular automata, and artificial life. An example of such a computational environment is presented here.

### **IMA: The Integrating Modeling Architecture**

Marcus Daniels <mgd@santafe.edu> (Santa Fe Institute)

TBA

### **Potential Of Ant Colony Optimization To Satellite Image Classification**

Raj P. Divakaran <rajprasanth@cc.usu.edu> (Department of Geography, College Of Natural Resources, Utah State University)

Studies on the behavior of social insects such as ants have shown that they exhibit a collective intelligence that cannot be explained based on the actions of the individuals that compose the colony. In an ant colony, global or emergent patterns emerge due to the local interactions of the ants. This ability to generate global patterns by a swarm of ants has served as the basis for the development of a set of algorithms that emphasize distributedness, robustness and flexibility - Ant Colony Optimization (ACO) Algorithms. Ant Colony Optimization techniques have been used to solve a number of network problems such as the Travelling Salesman's Problem and the Job Scheduling Problem. Vitorino Ramos and Filipe Almeida have shown that ACO algorithms could also be developed to solve the problem of image segmentation. This development is of immense use to researchers in the field of image classification and pattern recognition who have of late concentrated on developing robust, autonomous and flexible algorithms that can delineate the classes with minimum or zero inputs from the user. The ACO algorithms' potential for image classification has not been fully investigated so far. The purpose of this work is therefore to determine if ACO algorithms can be successfully applied to the problem of image classification of satellite imagery.

### **Classifier Systems in Swarm: a general purpose software.**

Gianluigi Ferraris <glf313@libero.it> (Torino, Italy)

Classifier systems are often used in AB models because they produce rules very closed to the problem and easy understandable. A matter in using CS is that the computation of the reward each rule has to be given may act dramatically on the learning capability of the method. Starting from the CW package I wrote in 1999 to handle CS, I've produced CW2 that includes optional features to handle rewards in a general way. CW2 removes the problem of reward computation and allows usage of both negative or positive amounts as well.

CW2 has been used in several models, two of them will be showed: the anteatr and the Bertrand's oligopoly. In the first model a CS is used by an anteatr to guess the behaviour of Langton's ants and eat them. Only observing ant's moves the CS reproduces the whole ant's rule set. In Bertrand's oligopoly the interplay between two enterprises is performed. Starting without any initial knowledge of the market players use own CS to make price-strategy, rules are evaluated by simply giving the profit/loss amount as reward. After learning the Nash's equilibrium is reached.

### **Recent Developments in Repast**

Tom Howe <thowe@flashfire.src.uchicago.edu> (Social Science Research Computing, University of Chicago) and Nick Collier (Social Science Research Computing, University of Chicago)

In addition to producing regular updates of our current behaviors, we have spent the past months developing the Recursive Porous Agent Simulation Toolkit (Repast) to include several new tools. One of the most significant change we have made to Repast's network support has been to restructure its approach to network visualization. In order to provide a more accurate view of social networks as they change, we have integrated network visualization techniques used in full-fledged network analysis packages, such as Pajak and Ucinet, into Repast.

We have also begun to integrate Geographical Information Systems (GIS) technologies into the Repast framework. At present, Repast has limited support for raster images. We intend to address this limitation by including vector type maps in the next release of Repast. The GIS support will include both visualization as well as limited geostatistical analysis capabilities.

Finally, we have released SimBuilder, a Rapid Application Development too, in beta form. At present SimBuilder only supports developing network models, but we are working to expand it to include spatial models as well. We have included NQPy, a Python-like development language, to simplify the development of models.

### **Top down and bottom up dynamic scheduling in Swarm: a comparison of implementations.**

Paul Johnson <pauljohn@ukans.edu> (Department of Political Science, University of Kansas)

The objective is to demonstrate the scheduling mechanisms available in Swarm and validate the implementation of models with decentralized, autonomous agent behavior. The example model in which these designs are investigated concerns public opinion dynamics and social networks. The model explores variants of Robert Axelrod's 'culture model'.

While all Swarm models are agent-based models, some models will orchestrate their actions from high level objects while others will allow the agents to 'schedule themselves' in a decentralized way. The Swarm class is vital because it controls the sequence of actions in the model. Swarms can be created at several levels so that the actions they imply are nested. Roughly speaking, there are two designs for agent-based simulation models. First, programs can be designed with 'top down' schedules, setups in which a higher level Swarm tells each agent to move (take a step). This approach is useful for traditional models of cellular automata, such as the Game of Life. These models can repeatedly iterate through the list of agents, either in a fixed or random order. The models are supposed to represent 'bottom up' processes, such as emergent properties, but the schedules themselves are imposed from the top. Some Swarm models that feature dynamic scheduling employ a variant of this top-down approach. Second, the design might be decentralized and dynamic. This is scheduling from the 'bottom up'. Each individual agent has its own scheduling apparatus. Each agent decides individually when it wants to act. This is especially meaningful if time is thought of as a series of days or years, each of which is composed of smaller time steps. The Swarm Activity library is designed to merge these individual actions into the overall time framework in a meaningful way. If agents schedule themselves to act at the same time, their actions can be executed in a first-come/first-serve manner or in a randomly sorted order.

### ***Using Java and Swarm to Build a Stochastic Simulation of Natural Organic Matter (NOM) and Microbe Interactions in the Soil***

Greg Madey <gmadey@nd.edu> (Computer Science and Engineering, College of Engineering, University of Notre Dame, Notre Dame)

Natural organic matter (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 ocean. NOM plays a vital role in ecological and biogeochemical processes, including chemical buffering, mineral dissolution/precipitation, photochemistry, and microbial nutrition. The production of NOM from biological precursor compounds and its remineralization into CO<sub>2</sub> are closely linked both to ecosystem function and to the global carbon cycle. No current models of NOM production and evolution, nor its interaction with microbes, describe both the quantitative aspects of organic carbon transfer and the qualitative aspects of NOM structural and functional heterogeneity.

We are partially funded on this project by a NSF ITR grant and are designing and implementing a stochastic simulation of NOM production and evolution using Java and Swarm. A preliminary design and prototype will be described. This is a joint project involving Stephen Cabaniss- Kent State University, Department of Chemistry, and Patricia Maurice- University of Notre Dame, Department of Civil Engineering and Geosciences.

### ***SMS: A Spinning Mill Simulator***

Matteo Morini <m.morini@labor-torino.it> (Centre for Employment Studies [<http://www.labor-torino.it/>], LABORatorio R. Revelli and CORIPE - Consortium for Research and Continuing Education in Economics [<http://www.coripe.unito.it/>], Università di Torino)

Production planning is a problem common to many industries: a timely delivery is often at odds with other technical and economical constraints.

In many cases even short period planning may be a big deal, even for medium to small-sized plants: it is not uncommon that many lots with different technical specs are to be interleaved in the production plan. Production units then need to be fine-tuned accordingly. Finding the most appropriate sequence is an astounding task, involving a huge number of combinations, which is nowadays performed by experience and common sense, essentially by hand. SMS tries to automatically sort out the best sequence combining a 1:1 model of a firm's supply chain (able to evaluate each solution's 'fitness') with a genetic algorithm (G. Ferraris' Genetic Manipulator v.0.2), both built in Swarm, which, instead of attempting a brute force approach bound to fail, searches the solutions' space in a clever way, often converging to a good solution in a matter of minutes on standard equipment.

### ***The Beer Dock: Three and a Half Implementations of the Beer Distribution Game***

Michael J. North <north@anl.gov> (Decision and Information Sciences Division, Complex Adaptive Systems Section, Argonne National Laboratory) and Charles M. Macal (Argonne National Laboratory)

The Beer Distribution Game is a classic supply chain problem widely used in graduate business programs to teach the concepts of supply chain management (Mosekilde, Larsen, and Sterman, 1991). It is well suited for this purpose since it is simple enough to be easily understandable but complex enough to be interesting. In particular, the Beer Distribution Game is notable for its ability to confound typical human players (Sterman, 1987, 1989). Many people who play the game find it difficult, if not impossible, to avoid the chaotic operating regimes that are the game's hallmark.

Docking or alignment is a verification process that seeks to find isomorphic relations between two or more implementations of an underlying model. Docking or alignment are used to "determine whether two models can produce the same re-

sults, which in turn is the basis for critical experiments and for tests of whether one model can subsume another" (Axtell, Axelrod, Epstein, and Cohen, 1996).

The original Beer Distribution Game was implemented as a systems dynamics model. The authors have implemented the game using Mathematica functional programming, RePast agent-based modeling and simulation (ABMS), and Swarm ABMS. The authors have reproduced all the published results with these new implementations. As part of this process, the authors have docked the implementations. The docking process was found to be challenging and time consuming, but ultimately rewarding. Several common themes of potential value to other researchers arose during the model docking process.

### ***Particle Economics Research Institute: the PartEcon Model***

Zoe-Vonna Palmrose <palmrose@marshall.usc.edu> (University of Southern California)

Particle economics describes a model and a methodology for generating emergent economic behavior driven by the creation of new modes of economic activity. An abstract agent-based model of generation of 'products' from 'input objects' provides the basic structure. These 'creation nodes' also keep track of their activities using the discipline of double-entry bookkeeping, monitoring continuously what they traded to whom, when, and for how much (of other products). This knowledge is used to update their beliefs about the values of products and the risk associated with transactions. There is no universally accepted 'money' to begin with. Nodes must decide how to measure their fitness as they make their way in their context of other nodes. This optimization of fitness as well as measure of fitness eventually leads to the emergence of money. In addition, the influx of new modes of economic activity generates Schumpeterian cascades of creation and extinction of whole communities of modes of economic activity. These cascades are described both in extent and time by power laws.

The PartEcon research team consists of:

- Mike Brown, retired CFO of Microsoft, past Chairman of the Nasdaq Stock Market, and past Trustee of the Financial Accounting Foundation.
- Jim Herriot, computer scientist, cognitive scientist, complexity scientist, previously with Sun Microsystems Laboratories, currently with BiosGroup, a complexity science think tank in Santa Fe.
- Zoe-Vonna Palmrose, Pricewaterhouse Coopers Auditing Professor at the University of Southern California.
- Bruce Sawhill, theoretical physicist, complexity scientist, previously with Stanford University, The Santa Fe Institute, currently with BiosGroup in Santa Fe.

### ***Modularity and Robustness in Organizations***

Paolo Patelli <paolo@santafe.edu> (Santa Fe Institute)

Manufacturing cells (modules) represent a possible new organizational form that shares analogies but also differences with past organizational solutions. Modules emerge at macro level as units performing manufacturing and non manufacturing activities; a micro level module combines different resources in a permanent way. The implications of a 'modular organization' are several: large organizations emulate and reinterpret market's mechanisms and the principles of near-decomposability and self-organization.

The aim of the model is to test different organizational structures against a given environment. An organization has the ability to process information from the environment and diffuses the results within the organization. It is structured as an hierarchy, every element of the organization has finite computation capabilities and the ability to adapt his behavior accordingly to a feedback signal. A modular organization is characterized by the level of redundancy i.e. by the number of duplicated and overlapped tasks. I have been studying the effect of environmental structural change on the behavior and performance of the firm.

### ***VSB 2.0: Visual Swarm Builder***

Alessandro Perrone <alex@unive.it> (Department of Economics, University of Venice)

This is an improved version of the Visual Swarm Builder that was already introduced during the 2001 SwarmFest. This is the new version and the first publicly available.

The need of a common language for agent-based economic simulations is, by now, widely recognized and several attempts are currently underway to identify the fundamental characteristics that such language should have. Between the two extremes of 'just add water' and 'do it yourself' several options have emerged in the last few years. Swarm, Ascape, Repast, Starlogo, AgentSheets and few other simulation platforms are, as of today, the most credible contenders.

VSF 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.

VSF is an integrated environment for the design and realization of Swarm code based on a menu driven selection of objects like schedule of events, graphics, lists and arrays etc. A visual building tool is clearly a useful extension to any programming environment, the more so for a large, complex and powerful simulation platform like Swarm. The absolute beginner as well as the expert programmer will find VSF useful. Because of its ease of use the former, and thanks to the speed with which a prototype can be obtained, the latter.

***More Fishy Business: Using Swarm Models to Test Theories of Fish Ecology*** [<http://math.humboldt.edu/~simsys/>]  
Steve Railsback <LRA@Northcoast.com> (Lang, Railsback & Associates and Humboldt State University)

Three recent studies used Swarm models to test theoretical assumptions of fish ecology. The first two studies used an agent-based model (ABM) that simulates how stream trout select their habitat considering how mortality risks, growth potential, and competition with other trout varies among locations. (1) Field studies show that when densities of juvenile trout are high, the average trout size is lower. These studies are used to support the theory of density-dependent growth: higher densities of fish produce more competition for food and therefore lower growth. Our ABM reproduced the observed decrease in fish size as density increased, but also found that growth rates were higher, not lower, when density was high. The decrease in fish size apparently was due to factors other than food competition, for example increasing rates of mortality in larger fish. (2) The models most commonly used to manage fish and wildlife habitat are based on the assumption that the habitat where animals are most commonly observed is the highest quality habitat. In the trout ABM, we know the intrinsic quality of each habitat patch and compared it to the number of fish using each patch in simulations. Even though fish in the ABM select the best available patch, our results contradict the assumption that the most commonly used habitat is the highest quality habitat. (3) Ocean herring migrate in large schools that have been observed to drastically shift destinations in years when the number of 'experienced' migrants is low compared to the number of juvenile herring that have never migrated before. Research shows that a fish school can be considered a self-organized aggregation of individuals following rules similar to those of Craig Reynolds' famous 'Boids' model. We used an ABM similar to Boids to test the theory that the 'experienced' migrants can control the direction of the fish school only if they are a sufficient fraction of the fish in the school. We found a sharp threshold in the ability of experienced fish to control school migration: as their fraction increased from 4% to 7% of the total, their control went from none to complete. These studies contribute to the debate over which is fishier: ABMs or conventional ecological theories.

***Can Agent-based Modeling Contribute to Scientific Theory?***

Steve Railsback <LRA@northcoast.com> (Lang, Railsback & Associates) and Paul Johnson <paul.john@ukans.edu> (University of Kansas)

Scientists that use agent-based models (ABMs) are often asked how their simulations can contribute to 'theory'. ABM simulations are sometimes viewed as too specific and parameter-dependent to lead to general theory. We will lead a discussion of such questions as: (1) What is 'theory' in the context of ABM simulation and how does it differ from classical theory? and (2) How can ABMs be used to develop theory? Examples from social and ecological sciences will be used to illustrate the discussion. In ecology, classical theory generally consists of equations assumed to represent system-level behaviors. In social sciences, theory often addresses how individuals interact. With ABMs, we start with the fundamental assumption that system-level behaviors emerge from the traits of individual agents and their environment. 'Theory' in the ABM context consists of models of agent traits: how agents interact with each other and their environment to produce emergent system properties. These models of agent traits may be specific to certain circumstances and agent types, or they may be quite general and widely useful. A key advantage of theory in an ABM context is that ABM-based theory is readily tested: ABMs can be used in a variety of ways to test the accuracy and generality of models of agent traits.

***Experimental Spatial Economics*** [<http://cc.usu.edu/~slk1r/technology/swarmfest2002.html>]

Yuya Sasaki <dirtyyuya@yahoo.co.jp> (Department of Geography And Earth Resources, Utah State University)

The consideration of geographical or spatial factors on economics is not a new trend. Many of those conventional approaches have tried to explain this relationship as a linear process, partly because of the difficulty to deal with its complexity and because of the lack of experimental tools. Under the hypothesis of non-linear process, however, the degree of the direct dependence of an economical entity (agent) on spatial factors is less than what is explained in linear process. Instead, an agent acts more in harmony with what other agents are doing in a common space. In the complex system of the real world, the initial influence of spatial factors on the determination of the economic activities is very trivial. But as time passes, this trivial influence will indirectly accumulate in the way that is called positive feedbacks or increasing returns. Thusly, it 'seems' that spatial factors solely have great magnetism in economics, while it is actually the entire system that is playing more important roles. By observing agent-based model, we can know in detail how a minor change in economic niche can greatly affect the spatial patterns through positive feedbacks. This paper introduces the concept of geographical

selection that is perceived in non-linear process, with the cellular automata (cell automaton) simulation model based on the Swarm development environment. The result shows that the tendency of spatial distribution of industry gets strengthened over time through the positive feedbacks, all other things being equal.

### ***Improved Java Virtual Enterprise (jVE) in Swarm***

Pietro Terna <pietro.terna@unito.it> (Department of Economic and Financial Sciences, University of Torino)

This is an improved version of the agent based enterprise simulator that was already introduced during the 2001 Swarm-Fest. Our simulated enterprise has both orders to accomplish - described by 'recipes' that contain what to do - and units that perform the different steps of production. Units can be within the firm or outside, constituting other complex enterprises or standing alone as small business actors.

The core of the model is now the clear separation between WD and DW, i.e. between the order recipes (WD: What to Do) and the operating units (DW: which is Doing What). WD and DW are completely independent, in formalism and in code. So, running the model, we can check the consistency of the two sides, as in the actual world, where the output of an enterprise arises from a complex interaction among products and production tools.

Recipes now deal with internal parallel production paths and include assembly steps, where the typical just in time problem of a supply chain can be tested. With the model we present five agent based simulations applied to enterprises. Finally, the theoretical goal of the research is illustrated: the reproduction of the trial and error process that generates new enterprises.