*Proceedings of the 2004 Winter Simulation Conference*
*R .G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters, eds.*

# SIMULATING GROWTH DYNAMICS IN COMPLEX ADAPTIVE SUPPLY NETWORKS

Surya Dev Pathak
David M Dilts

Management of Technology Program (EECS)
Vanderbilt University
Box 1518, Station B,
Nashville, TN-37235, U.S.A.

Gautam Biswas

Department of Electrical Engineering and Computer Science (EECS)
Vanderbilt University
Box 1824, Station B
Nashville, TN-37235, U.S.A.

## ABSTRACT

This paper discusses an extended adaptive supply network simulation model that explicitly captures growth (in terms of change in size over time, and birth and death) based on Utterback's (Utterback 1994) industrial growth model. The paper discusses the detailed behavioral modeling of the key components in the model with the help of statechart and decision tree representations. The design of a distributed, multi-paradigm, agent-based simulation that addresses the issue of scalability and computational efficiency is presented. The system is targeted to run on a supercomputing grid infrastructure at Vanderbilt University. We present a method for validating this model using an experimental design that models the growth dynamics of the US automobile industry supply network over the past 80 years. The experimental work is now in progress and the results and analysis of this work will be presented during the conference.

## 1 INTRODUCTION

How do supply networks grow and emerge? Are there ways for foreseeing the implications of current policies on the future evolution of supply networks? To answer these questions, it's important not to look at supply networks as mere dynamic flow networks with a fixed structure but as dynamic systems whose structures evolve and change (Harland et.al 2002; Choi, Dooley, and Rungtusanatham 2001). In our previous research (Pathak and Dilts 2002; Dilts and Pathak 2003; Pathak and Dilts 2003; Pathak, Dilts and Biswas 2003; Pathak and Dilts 2004) we modeled the structural and behavioral dynamics of supply networks as complex adaptive systems and used multi-paradigm agent based simulations to demonstrate emergence, perturbation effects and sensitivity to initial conditions. The survival of firms was linked to a fitness function and firms whose fitness fell below a threshold died. Previous work only partially modeled the growth phenomenon in supply networks; however, birth of new firms was not explicitly included in the model. This new work explicitly models the growth dynamics and evolution in supply networks by extending the industrial growth model of Utterback (Utterback and Suarez 1993; Utterback 1994), and conceptualizing network emergence based upon a predictable growth pattern. We propose that the emergence patterns that supply networks will follow are similar to the "bell-shaped curve" discovered by Utterback (1994).

Our basic simulation model breaks down a complex adaptive supply network into two principle components, namely, (i) the supply network environment, and (ii) the firms involved (represented as nodes in the network model). The nodes operate using simple decision-making rules and attempt to satisfy the environmental demand. As new industries are created and supply networks start emerging, new firms may enter the market (birth). Some firms are successful and form relationships (grow). But some firms fail (die) either due to local conditions or because they cannot become part of a viable supply network. Thus, over time, supply networks may grow into relatively stable structures based on the interactive effects of the local decision making rules and the environmental factors (Choi and Hong 2002).

To study the emergence of networks that incorporate both birth and death patterns, we have redesigned CAESAR (Pathak, Dilts and Biswas 2003; Pathak and Dilts 2004) and significantly enhanced the behavioral complexity of the fundamental entities of the model: environment and the node. Section 2 of this paper presents the former complex adaptive supply network model and its limitations. It then introduces Utterback's theory and the notion of growth in supply networks. Section 3 presents the simulation design based on an updated model. Section 4 presents the behavioral representations for the fundamental entities of our model, where we show how the current design addresses the scalability and synchronization issues that were a problem with our previous model. Section 5 briefly introduces the advanced version of CAESAR called CAS-SIM (Complex Adaptive Supply network simulator), a distributed agent based tool suite for implementing the simulation model. Section 6 then presents the experimental design for simulating the US automobile industry growth curve (testing proposition), followed by section 7, which discusses the expected results from this experiment and the proposed analysis. The current work is ongoing and Section 8 summarizes progress so far and presents the future plans.

## 2 BACKGROUND: COMPLEX ADAPTIVE SUPPLY NETWORKS

### 2.1 Conceptual Research Model

Pathak, Dilts and Biswas (2003), provide description of various approaches used by researchers for analyzing supply networks. We conceptualize a supply network (Figure 1) as a system consisting of an environment; the market, in which firms (nodes) reside and interact based on simple behavioral rules to fulfill global demand. Stochastic environmental parameters describing market conditions and demand, the decision-making scheme employed in the nodes, and the differential fitness function used to model node strength, all combine to influence the structural as well as behavioral dynamics of the evolving supply network.

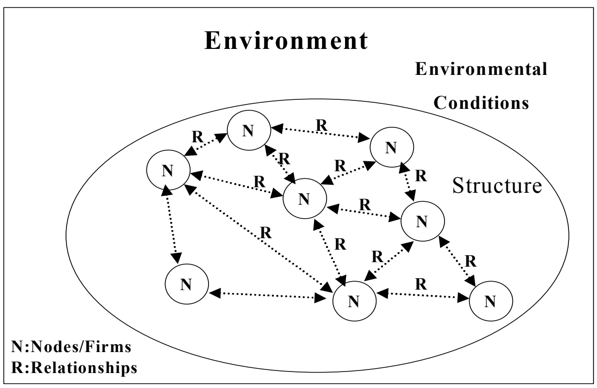


Figure 1: Conceptual Research Model

A complex adaptive system (CAS) approach is well suited for modeling systems with such structural and behavioral dynamics. In such systems, the networks emerge over time into a coherent form, adapting and organizing themselves without any singular entity controlling or managing the global structure or node interactions (Holland 1996). CAS is characterized by three major components; namely, 1) the *Environment* that the network resides, 2) *Internal mechanisms* (dealing with *agents, schemas, connectivity* and *dimensionality*), and 3) *Co-evolution* (quasi equilibrium and state changes, non-linear changes, and non-random future) (Choi, Dooley, and Rungtusantham 2001). We use similar constructs to define a high level model of the CAS approach (Figure 2).

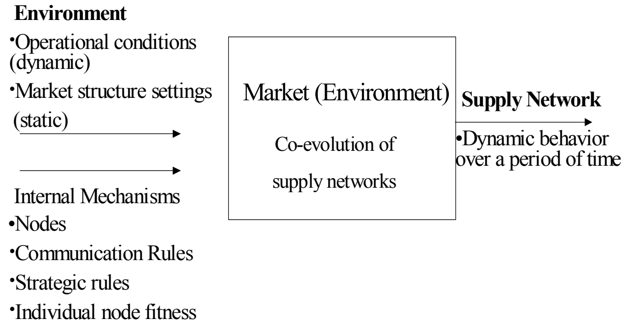


Figure 2: High Level Model

1. *Environment:* This is where the supply network entities (nodes) reside. The environment is characterized by two major conditions: (i) Operational conditions, which specify the demand, timing and cost related information, and (ii) Market structure settings, which specify government regulation and policies, or business rules imposed on the system. For example, if we consider the automobile industry supply network, product demand and its timing are operational conditions while the domestic automobile market and the government's CAFE (fuel efficiency) standards are part of the market structure settings. The environment also sets a fitness threshold, which is the minimum fitness necessary for a node to survive in the environment.
2. *Internal Mechanisms:* Nodes (*agents/firms/entities*) represent firms in a supply network. They are goal driven. Every node has a pool of strategies to use in achieving local goals. Rules (*schemes*) operationalize these strategies and are driven by objectives that define the goals of a node and constraints that are imposed by the environment and the node itself. An example of a simple objective for a node is to be a low cost producer while not to interacting with more than one high level node in a supply chain. Responsiveness, capacity and budgetary constraints are examples of additional node constraints. Generally, nodes make two types of decisions, (i) who to work with in the environment (partly driven by market rules), and (ii) how to strategically determine decisions such as capacity, degree of outsourcing, and price of the product when accepting an order. Continuing with the automobile industry example, nodes (manufacturers and suppliers) are driven by internal strategies/policies, such as manufacturing strategy, production policy, supplier selection policy and pricing policies. These policies dictate the firm's behavior in the supply network and result in *connectivity* between nodes. Nodes have a fair degree of autonomy (*dimensionality*) in selecting strategies. Decision-making rules used in this re-

search are based on widely accepted market structure (Shy 1996) and game theory literature (Rubienstein and Osborne 1994).

3. *Co-evolution*: The results of implementing node strategies in specific environments, generate *co-evolving* supply networks structures. Co-evolution is an output, that is, it is the result of the interaction between the environment in which the supply network exists and the internal mechanisms used by the nodes in the network. The co-evolution process results in a differential growth of individual firms due to the inbuilt fitness functions (Internal mechanisms) and may result in death of weak firms.

The model just presented has a limited notion of growth and evolution of the network over time. For a supply network to be truly emergent, a model should capture the possibility of dynamic birth of firms as the co-evolution process unfolds. The next section presents the Utterback's industrial growth model (1993, 1994), which we have extended to model dynamic growth in supply networks.

### 2.2 Capturing Growth Dynamics: Utterback's Model

As new industries emerge, supply networks grow along with them with new relationships being formed between firms that work collectively to meet demand. Utterback's model of industrial growth considers that at the inception of an industry, the entry barriers for firms are low and there is no clearly defined market structure. At this stage there are a number of new entrants, with each firm trying to establish itself as a leader. The next phase in Utterback's model consists of the establishment of a clearly defined market structure with firms focusing more on economies of scale and network externalities. Not all firms are successful, and the unsuccessful firms are "shaken" out of the market. As time progresses the number of new firms entering the market declines. The different phases of the industrial growth cycle are thus represented by a bell shaped curve (Figure 3).

Utterback also suggests that firms learn to play specialized roles over time. Thus in the beginning of an industry all firms are generalists. This can be seen if we look back at the automobile industry market in the US in the early 1900's. At that time "buggy" and "bicycle" makers were trying to make cars. But as time progressed only a few "generalist" firms (i.e., assemblers) remained, and remainder of the firms either died or learned to play a "specific" supplier role.

Utterback's model captures growth of an industry only with respect to the number of firms entering and exiting the market (Utterback and Suarez 1995). But another dimension that needs to be captured is the "size of the firms". In one possible scenario, as market size increases with time, there is a differential growth in firms with some firms

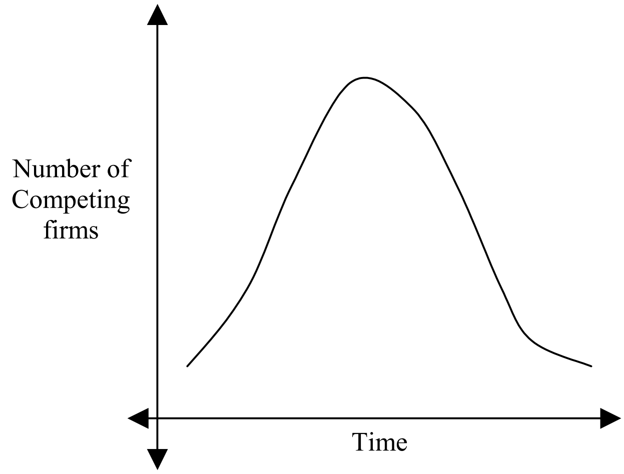


Figure 3: Utterback's Industrial Growth Model

expanding capacity greatly, while others fail. An alternative scenario might be where no set of firms dominate so the market gets divided relatively equally between participating firms. An example of the first scenario is the automobile industry, and the second is the pharmaceutical industry. Thus, in our complex adaptive supply network model growth is considered from two perspectives: number of firms and size of firms.

## 3 SIMULATION MODELING

To understand the growth and evolution dynamics we need to observe the time dependent behavior of the model just discussed. Simulation is a widely accepted methodology for studying time varying properties of a system (Ziegler, Praehofer, and Kim 2000).

### 3.1 Simulation Model Architecture

In our prior research (Pathak, Dilts and Biswas 2003) we used a multi-paradigm architecture (Ziegler, Praehofer, and Kim 2000) as shown in Figure 4 to build the simulator. In the multi-paradigm architecture, some of the components of the model, such as the environment fit a discrete time modeling (DTS) paradigm. On the other hand individual node behavior was event driven, and best captured by a discrete event formalism (DEVS). In the old architecture the environment and the evaluator (which acts as a controller for the environment) are coupled models as they interact with multiple nodes in the system.

As shown in Figure 4, the Environment agent acts as the root coordinator and is a coupled model. Evaluator, Visual Manager and Timekeeper are children, which the environment launches and controls. The environment runs on a simulated clock. **Evaluator** acts as a coupled DEVS coordinator as it owns all the nodes and communicates with them using a message passing protocol. The evaluator launches all the nodes and sends them demand information and other messages. At the end of a fixed number of demand cycles it also evaluates all the nodes and kills the unfit ones. **Nodes** are atomic DEVS models and are owned and coordinated by the evaluator. The detailed working of each component is described in Section 4, behavioral modeling.

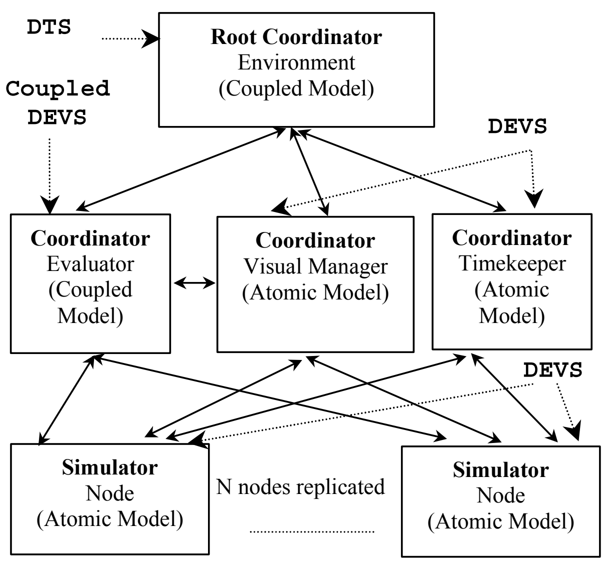


Figure 4: Multi-Paradigm Simulator Architecture

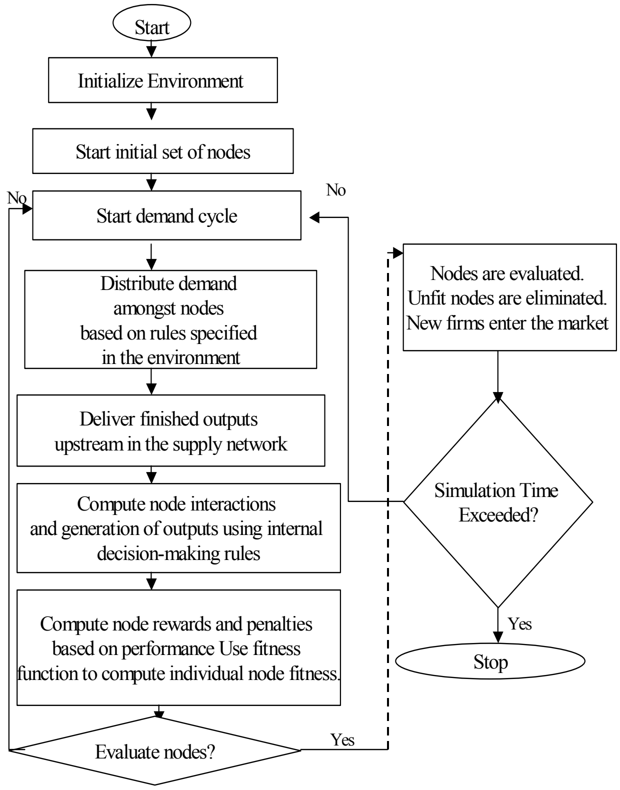


Figure 5: Typical Simulation Run

## 3.2 Simulation Algorithm

Figure 5 captures the sequence of events in a cycle of the simulation algorithm that leads to the emergence of a supply network. The simulation begins with the environment initializing itself and setting the external system parameters such as the start time of the simulation clock, creating a demand function, activating the evaluator component, and assigning values to all other operational conditions. After this initialization period, an initial number of nodes are generated (birth).

The environment then starts a new demand cycle and the evaluator distributes the demand between all the nodes based on the market structure settings specified in the environment. The nodes interact amongst themselves driven by their internal mechanisms to fulfill the period's demand. Finished goods are delivered upstream from subcontracting nodes to all the way up to the final customer. After profits and losses are calculated for individual nodes, each node updates its current fitness value based on its specific fitness function. The evaluator periodically checks the fitness of all nodes in the current population and removes unfit ones, i.e., those that have fallen below the environmental fitness threshold. Depending on the demand supply curve new nodes are inducted into the environment (dynamic growth).

The number of simulation demand cycles is set during the environment initialization process. If the current demand continues generating demand. cycle has not exceeded this number then the environment

As is evident from this description, the notion of time is made explicit in the supply network by making the demand cycle occur at regular time intervals. The demand generation is asynchronous with respect to a node. A node is implemented such that it is capable of handling multiple demands. The only time nodes and the environment need to be synchronized are when the evaluator evaluates unfit nodes. Individual node behaviors and their interactions with other nodes are event driven. This results in a multi-paradigm discrete time and discrete event (Cassandras 1993; Ziegler, Praehofer, and Kim 2000) computational model.

# 4 BEHAVIORAL MODELING OF KEY COMPONENTS

The detailed behavior of some of the components in the simulation model can be shown using a state chart representation. The environment in the model is used for defining the market settings and generating demand patterns and hence has a simple state chart representation as shown in Figure 6.

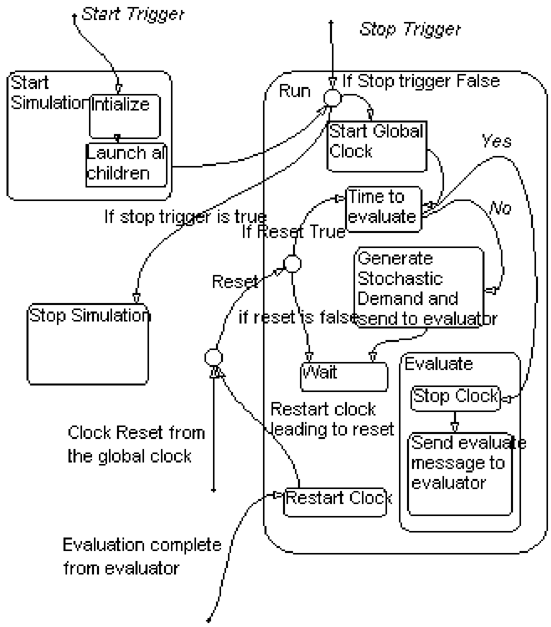


Figure 6: State Chart Representation of Environment

The environment is the first to start in the simulation and initializes itself. It then launches the evaluator, the support agents (visual manager and timekeeper) and goes to its run state. In its run state it first starts the global clock and transitions to the next state to check if it is time to evaluate for unfit nodes. In the first run, the flag is not set and the environment generates a stochastic demand and sends it to the evaluator. The environment keeps generating these demands asynchronously. The evaluator and the nodes have a multi-threaded implementation to enable parallel processing of these demands. This corrects the earlier problems the earlier model had with synchronization and inefficient simulations due to tying up of the nodes to the global clock. When the "time to evaluate" flag is set, the environment stops the global clock and requests the evaluator to evaluate all the unfit nodes.

Figure 7 shows the state chart representation for the evaluator. The evaluator, upon receiving the launch message, goes into the Start state, initializes itself, and launches the initial number of nodes set in the environment. It then waits for a demand from the environment. Once it gets the first demand it goes into the run state and distributes demand based on the game theoretic and market structure rules. It then returns and waits for further demand messages from the environment. When it receives a evaluate message from the environment, it first broadcasts a pause message to all nodes so that it can flag all the unfit nodes that are below the environmental threshold level, such that they cannot get any new orders. It then removes all the nodes from the simulation that have been previously flagged. The evaluator also responds to other messages from nodes, such as the node fitness report, and so on.

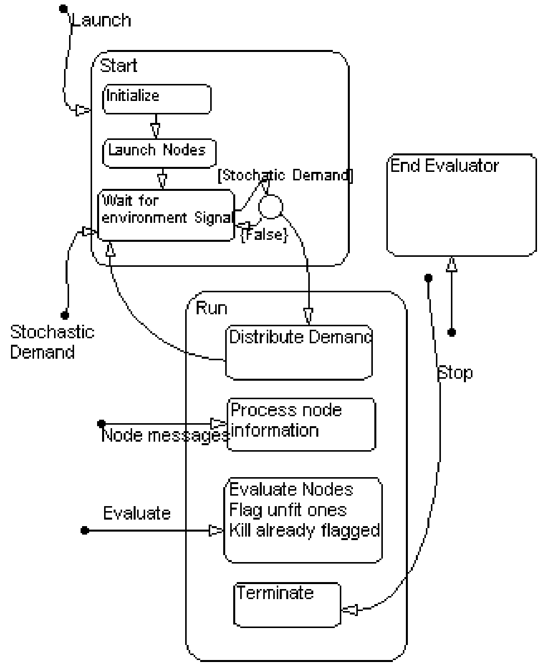


Figure 7: State Chart Representation of Evaluator

Figure 8 represents a high-level state chart representation for node behavior. A node is completely event driven. When it is first launched by the evaluator, it enters its start state and initializes itself. It then transitions to its run state. In its run state it waits for incoming messages. A node fundamentally responds to seven basic messages or events. We use decision tree diagrams to illustrate node behavior in response to some of the important messages.

1. Pause. After every 12 demand cycles (equivalent to a "month" in simulated time), the evaluator evaluates the nodes. During this time no transactions take place in the environment and to facilitate this the evaluator sends a "Pause" message to all nodes. Upon receiving this message a node suspends all its activities.
2. Report. Node responds to this message by sending back its current fitness value to the evaluator.
3. Flag. This message sets the death flag in a node. This tells the node that it will be removed from the environment in the next cycle and it does not get any new orders. The node cannot control this flag.
4. Demand. Figure 9 describes the decision tree representing the behavior of the Fulfill demand state that is triggered due to this event. Once a node is awarded an order, it sends out a RFP (Request for Proposal) and waits for a fixed amount of time for the bids to arrive. It then compares its internal assembly cost with respect to the sub-contracting cost. If the internal cost is lower and the demand is

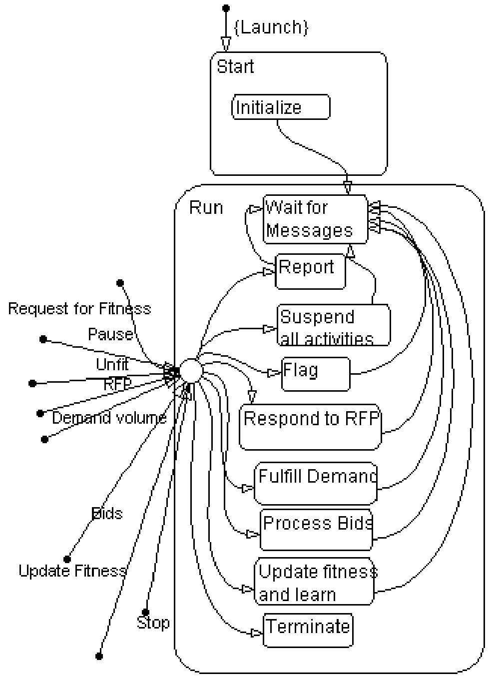


Figure 8: State Chart Representation of Nodes

less than the current capacity then the entire demand is manufactured and shipped. If the demand is greater then node capacity, it can either decide to accept the penalty of not meeting the demand and produce up to capacity or else undergo a temporary expansion, especially if it improves the profit margin. If the node subcontracts, then as described in case of the evaluator it follows Edgeworth's version of Bertrand's pricing game (1925) and distributes the demand between the responding bidders.

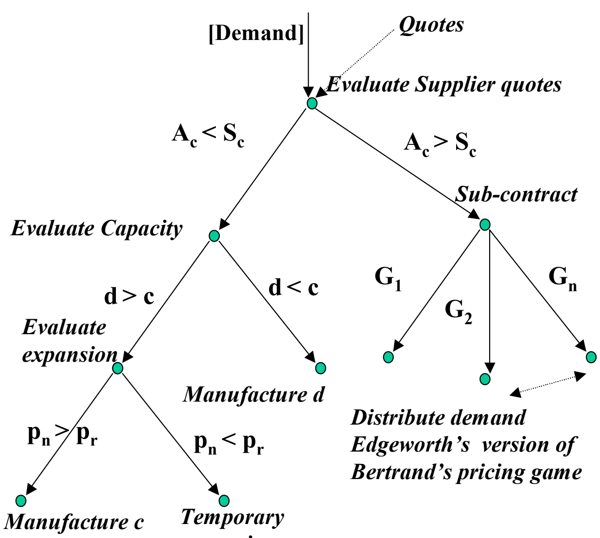


Figure 9: Decision Tree Embedded in the Fulfill Demand State

5. Request for Proposal (RFP). When a node receives a request for proposal, it responds based on its role propensities. Every role a node can play has an associated propensity value. Every node also has an associated "available to promise" (ATP) capacity by role. BidDing is based partially on role propensity and role ATP. If a node receives a RFP (due to a new demand in the environment) while it is still processing a current demand, it uses the ATP capacity to bid on the new demand and thus tries to ensure that it doesn't remain idle in the near future (see Figure 10).

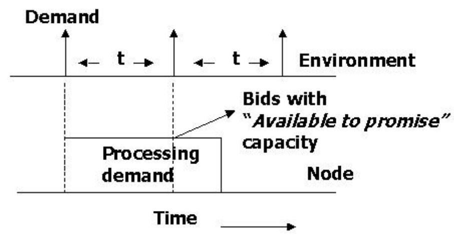


Figure 10: Parallel Response to Multiple Demands

6. Time. A node requests a separate timekeeper agent for waiting on bids. When time is up the node receives this message.
7. Update Fitness. Figure 11 represents the Learn state behavior. Upon receiving this message, a node activates its learning module and adapts its behavior according to its performance in the current demand cycle. If it results in a positive change in fitness then it updates the propensity of playing that role. It then checks if the immediate history of demand cycles (number of demand cycles are heuristically fixed) has yielded a positive growth. If yes, then it expands its current capacity under that role, else it stays at the current capacity. If it experienced a negative fitness growth then it decreases the propensity of playing the role and checks if the immediate history of demand cycles has yielded negative growth. If yes then it shrinks its current capacity associated with that role, else it remains at its current level. At the end of both of these growth cycles a node updates the probability of playing its current pricing strategy once again. If the change in fitness (δf) is greater than the aspiration level (a), then it increases the probability, else decreases it. The aspiration level indicates what a node thinks is a successful outcome. From time to time a node excites (modifies) the aspiration level, so as to experiment around the strategy space (Karandikar et al. 1998).

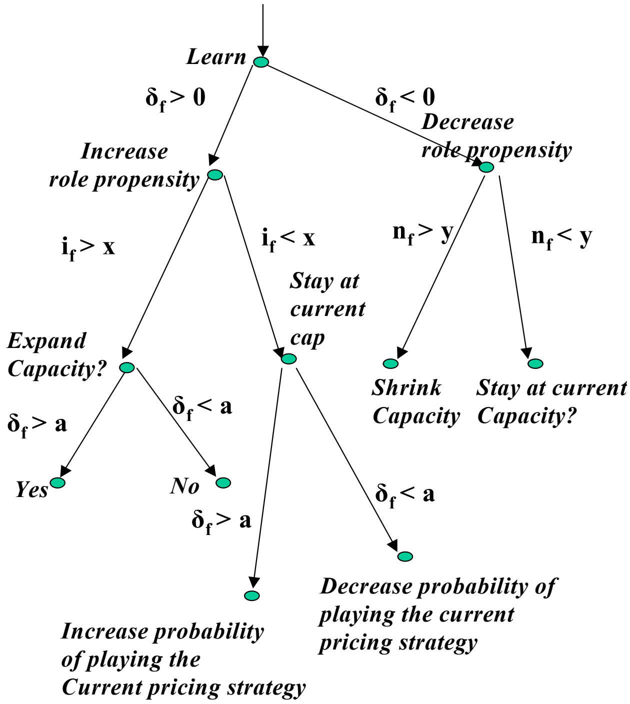


Figure 11: Decision Tree Embedded in the Learning State

## 5 IMPLEMENTATION

To implement the advanced multi-paradigm simulator, we have developed a tool suite called CAS-SIM (Complex Adaptive Supply Networks Simulator) (Pathak and Dilts 2004). This suite is built using multi-agent-based techniques (Ferber 1999) to capture dynamic interactions between nodes and the changing configuration of the network for each demand cycle. The principal entities in the simulation model such as the environment, evaluator and the nodes are represented as software agents with built in behavioral rules. CAS-SIM uses MadKit (a Java based agent package) as an agent platform.

The main issue with the previous version of CAS-SIM was scalability. The prior design would not allow the scaling up to a large number of nodes on a single processor. To address this issue, the current version of CAS-SIM is designed such that it can model agents and distribute them over multiple processors. Using such a "hub and spoke model", the environment and the support agents (visual manager, timekeeper) are run on the hub and the nodes representing firms are distributed on the "spokes". The Vanderbilt University's grid computing infrastructure called "Vampire" is being used as the computational platform for executing the parallel, distributed simulations. By distributing the agents on this high performance grid, we hope to significantly increase the scale of future simulations.

## 6 VALIDATION EXPERIMENT DESIGN

Utterback (1994) has recorded the growth phenomenon of numerous 20th century industries in the US such as the automobile industry, television, and typewriter. For investigating the growth phenomenon in supply networks we have selected the US automobile industry in the 20th century. In the beginning of the century there were about 100 automobile manufacturers (Utterback 1994). The entry barrier to the car market was low and the market itself was not clearly defined. Over time some firms developed special roles in the form of assemblers (GM, Ford) and some developed supplier roles. Today there are few major domestic automobile manufacturers in US, but a large number of supplier firms organized in a multi-level tier supply network structure. The automobile market grew into a very deeply hierarchical structure over time. Based on this example and using known automotive industry structure parameters, we will use the simulation runs from the model to establish the following propositions:

*Proposition 1: A growing demand curve will yield an Utterback-type supply network growth curve (bell shaped curve).*

*Proposition 2: A growing demand curve will result in a supply network structure that matches the deep hierarchical structure of the current US automobile industry.*

### 6.1 Setting Up the Experiments

To test these propositions we have designed an experiment with the following setup:

The primary product is a passenger car with 5 subparts (simplification)

a. Demand. Demand is set up as a Gaussian distribution N ~[mean, std dev], where the mean value (the number of automobiles manufactured) is derived from the automobile industry data (Ward's automotive report 2002) and the standard deviation is set to 5% of the mean value (initial assumption). We have collected known demand data for passengers cars in US from 1920 to the current time. For example, in the year 1950 the demand was 6,628,598 units. Thus in our simulation we use a demand curve N~[6628598, 331430 ].

b. Demand cycles and evaluation cycles. Experiments will be run for 960 demand cycles and evaluation of nodes will take place periodically after 12 demand cycles. If we consider each cycle to be a month, the each node is evaluated after a year of operation. This is typical in actual firms. The total number of cycles is based on the average time it takes for new businesses to fail/succeed as reported in the new venture creation literature (Timmons 1999).

c. Fitness Threshold. Fitness threshold for the nodes to survive in the environment is set to 0.25. This value is assigned empirically and is not changed in the simulation run.
d. Initial Node Capacity. Initial capacity of a node is selected from a uniform random distribution in the interval [a, b]. "a" is set to a minimum level and b is calculated based on the initial number of nodes and the mean demand to start with.

*b = mean demand /initial number of nodes*

For example, a is set to 10% of b. Initial number of nodes to begin with is 100 and the average demand for the year 1930 (from the Ward's report, 2002) is 2784745. Then

$$b = 2784745/100 = 27847$$
$$a = 0.1*b = 2785$$

The simulation starts with about 100 nodes. The figure is taken from Utterback's work (1994) on the US automobile industry, which indicates that there were about 100 large/ small automobile manufacturers in the beginning of the 20$^{th}$ century.

We are Primarily interested in the growth and structural evolution of supply networks. For commenting on growth we are planning to record node mortality, the demand profile and the node capacities over time. The node mortality will help in investigating the growth pattern in supply networks, where as node capacities and demand profile will help answer the question on growth of supply networks with respect to the size of firms and the market size. The relationships developed during the simulation will show the type of structure developed.

At least 100 samples are to be collected for the current experimental setup as such a sample size allows for statistically significant conclusions.

## 7 EXPECTED ANALYSIS AND RESULTS

We will perform repeated-measures time series analysis (Williams 1997) to identify the complex dynamics with respect to the output parameters defined occurring in a supply network system. This will strengthen earlier results (Pathak, Dilts and Biswas 2003) on the complex adaptive properties of supply network systems by specifically looking for perturbation effects, attractors and limit cycles. We expect to see a bell shaped growth curve over time. Also,

1. We expect to validate our research model, by showing that the simulation output matches the current automobile industry structure.
2. Based on the research framework, other supply networks can be similarly modeled and their behavior can be analyzed.
3. By investigating the effect of individual parameters on the supply network behavior, scenario analysis can be performed and thus the model has a predictive value that can aid in the decision making process in supply networks.

## 8 SUMMARY AND FUTURE WORK

The study of growth dynamics phenomenon in supply networks will hopefully provide insight on how different supply networks grow and emerge into a wide range of structures. In this paper, we have presented the detailed design of our growth oriented simulation model for supply networks. This extends our previous multi-paradigm simulation model, where the growth model was incomplete. A novel parallel distributed agent-simulation approach has been adopted for operationalizing the simulation model. For validation purposes we have designed an experiment simulating the US automobile industry over the last 80 years. These experiments will be completed in the immediate future and the final results will be present at the conference. In addition to running this initial experiment, we are in the process of characterizing other industries and observe if the framework can simulate these industries as well.

## 9 ACKNOWLEDGMENTS

We would like to express our sincere thanks to Dr. Alan Tackett, from the Vampire group at Vanderbilt University for his support and help in setting up the experiments on the Vampire cluster.

## REFERENCES

Cassandras, C.G., "Discrete Event Systems: Modeling and performance analysis". Richard Irwin, NY, 1993.

Choi, Y.T., Dooley, K.J., and Rungtusanatham, M., "Supply networks and complex adaptive systems: control versus emergence". Journal of Operations Management, 19: 351-366, 2001.

Choi, Y.T., Hong, Y., "Unveiling the Structure of Supply Networks: Case Studies in Honda, Acura, and DaimlerChrysler"., Journal of Operations Management, 20, 5, pp. 469-493, 2002.

Dilts, D.M., and Pathak, S.D., "Emergent Supply Chain networks". Presented at the Decision Sciences Institute Annual Meeting, Washington DC, 2003.

Edgeworth, F., "The pure theory of monopoly". In papers related to Political volume. London: Macmillan, 1925.

Ferber, J., "Multi-agent Systems," pp.10-11, Addison-Wesley Longman, 1999.

Harland, C.M., Lamming, R.C., Zheng, J and Johnsen, T.E, “A taxonomy of supply networks. IEEE Engineering Management Review,” 30(4): 2002.

Holland, J.H., “Hidden Order”. Reading, MA, Addison Wesley, 1996.

Karandikar, R., Mookherjee D., Ray, D., and Vega-Redondo, F., “Evolving Aspirations and Cooperation”. Journal of Economic Theory. 80.. 292- 331, 1998.

Pathak, S.D., and Dilts, D.M., “Simulating supply chains using complex adaptive systems theory”. IEEE International Engineering Management Conference, Cambridge United Kingdom, 2002.

Pathak, S.D., and Dilts, D.M., “Origin and evolution of supply networks using complex adaptive systems theory”, Production and Operations Management Society national conference, Savannah, Georgia, 2003.

Pathak, S.D., and Dilts, D.M., “CAS-SIM: Complex Adaptive Supply Network Simulator, A Scenario Analysis Tool For Analyzing Supply Networks”, Production and Operations Management Society conference, Cancun, Mexico, 2004.

Pathak, S.D., Dilts, D.M., and Biswas G, “Understanding the evolution dynamics of supply networks”. IEEE Engineering management Transactions (Submitted), 2003.

Rubinstein, A., and Osborne, M.J., “A Course in Game Theory”. MIT press, 1994.

Shy, O, “Industrial Organization: Theory and Applications”, The MIT Press, 1996.

Suarez, F.F., and Utterback J.M, “Dominant designs and the survival of firms”, Strategic management journal, 16 (6): 415-430 1995.

Timmons, J.A., “New Venture Creation: Entrepreneurship for the 21st Century” Irwin McGraw Hill, 1999.

Utterback, J., “Mastering the dynamics of innovation”, Harvard Business School Press, 1994.

Utterback, J.M., and Suarez, F.F, “Innovation, competition, and industry structure”, Research policy, 22 (1): 1-21 1993.

“Ward’s automotive reports” as published in Ward’s Motor Vehicle Facts and Figures, 2002.

Williams, G.P., "Chaos Theory Tamed"., 1st edition, Taylor and Francis, 1997.

Ziegler, B.P., Praehofer, H., and Kim, T.G., “Theory of Modeling and Simulation”, 2nd edition, Academic Press, 2000.

## AUTHOR BIOGRAPHIES

**SURYA D PATHAK** is a is graduate student in the management of technology program (Department of Electrical and computer science) at Vanderbilt University, working towards his PhD. His research interest lie, in the field of simulation of complex adaptive systems using Software agents. He is currently trying to come up with a fundamental theory behind formation of different supply chain structures. He can be contacted at `<surya.d.pathak@vanderbilt.edu>`.

**DAVID M DILTS** holds a joint professorship between the Owen Graduate School of Management and the Vanderbilt University School of Engineering. His areas of research focus on in the intersection of technology, management, and operations management. Currently he has an active research program that is studying such topics as Supply Chain Management, Extended Enterprise Integration, and the ability of information technology to radically improve the practice of medicine. He is on the editorial board of the Journal of Operations Management and the Decision Sciences Journal of Innovative Education.

**GAUTAM BISWAS** is an Associate Professor of Computer Science and Engineering, and Management of Technology at Vanderbilt University and a senior research scientist at the Institute for Software Integrated Systems (ISIS) in the School of Engineering. Prof. Biswas conducts research in Intelligent Systems with primary interests in hybrid modeling, simulation, and analysis of complex embedded systems, and their applications to diagnosis and fault-adaptive control. He is also involved in developing simulation-based environments for learning and instruction, decision-theoretic planning and scheduling techniques for intelligent manufacturing systems. Dr. Biswas is an associate editor of the IEEE Transactions on Systems, Man, and Cybernetics, the IEEE Transactions on Knowledge and Data Engineering, and the Journal of Applied Intelligence.