Fall 2007

# Architecting system of systems: artificial life analysis of financial market behavior

Nil Hande Ergin



Department: Engineering Management and Systems Engineering

ARCHITECTING SYSTEM OF SYSTEMS: ARTIFICIAL LIFE ANALYSIS OF FINANCIAL MARKET BEHAVIOR

by

NIL HANDE ERGIN

A DISSERTATION

Presented to the Faculty of the Graduate School of the

UNIVERSITY OF MISSOURI-ROLLA

In Partial Fulfillment of the Requirements for the Degree

DOCTOR OF PHILOSOPHY

in

SYSTEMS ENGINEERING

2007

| | |
|---|---|
| C. H. Dagli, Co-Advisor | D. Enke, Co-Advisor |
| S. Ramakrishnan | S. Grasman |

D. Tauritz

## ABSTRACT

Today's systems typically do not stand alone in isolation. Often a system fits within a System of Systems, a network of interconnected systems that often exhibits unpredictable behavior. This study is motivated by the challenges of understanding the emergent system level behavior of System of Systems given the opaque characteristics of social processes and continuously changing operating and environmental conditions. An artificial life based framework for modeling System of Systems is presented as an analysis technique. The framework comprises cognitive architectures embedded in multi-agent models. Financial markets are selected as an analysis domain to demonstrate the framework since they are a good example of self-organizing systems that exhibit System of Systems characteristics, specifically emergence on a grand scale. The effects of different mechanisms on system level market dynamics are analyzed. In particular, the effects of the covering mechanism, learning mechanism and bias mechanism are analyzed. A trader-based architecture is proposed to formulate a trader decision model that combines bias mechanisms with learning mechanisms. A prediction accuracy based Learning Classifier System is used to model the trader learning mechanism. Markov processes are utilized to model the bias mechanism of traders. Simulation experiments are generated using the Anylogic5.1™ software. Homogenous rational expectations equilibrium is utilized as the benchmark for comparison of results from the hybrid proposed model. The model derived from the framework contributes to understanding the market behavior and potential sources of deviation from efficient market equilibrium. The artificial life based framework provides a flexible way of modeling sub-systems of System of Systems and captures the adaptive and emergent behavior of the system.

## ACKNOWLEDGMENTS

I would like to thank my advisors, Dr. Cihan H. Dagli and Dr. David Enke, for their continuous support, guidance and friendship throughout this study. Their vision, enthusiasm and dedication to science and education have been influential. There are no words to express my gratitude, admiration and affection for them. It has been a privilege to work under their supervision. I would like to thank my committee member, Dr. Sreeram Ramakrishnan, for his guidance and feedback at critical points in my research. I would like to thank my other committee members, Dr. Scott Grasman and Dr. Daniel Tauritz, for their valuable suggestions throughout this study. All together my committee's insights enriched my dissertation immensely.

I would like to thank the Engineering Management and Systems Engineering Department for providing me funding during my study. I have been fortunate to work among fellow students in the Smart Engineering Systems Lab and the Laboratory for Investment and Financial Engineering whom I found to be intellectually stimulating. I would like to thank all my friends for their friendship, the list of their names will simply not fit in this page.

I am grateful and indebted to my mother and father, Yildiz and Erkan Kilicay, for their love, sacrifices and contributions throughout my life. I am grateful to my brother, Ozgur Kilicay, for supporting me and standing by my side throughout my tough days. Of course I would be remiss if I did not mention the extensive support and love of my mother and father-in-law, Sabahat and Ali Ergin, as well as my brother-in- law, Taner Ergin. I would also like to remember in memory of my grandfather, Dr. Ismet Gunes, whose life story and his dedication to medical science has been an inspiration on me.

Finally and most specially, this study is dedicated to my husband, Okan Ergin, who finds the best in me and stands by me with his endless patience, strength, friendship and love through the life journey.

# TABLE OF CONTENTS

Page

## LIST OF ILLUSTRATIONS

## LIST OF TABLES

# NOMENCLATURE

| Symbol | Description |
|---|---|
| W | Wealth of trader |
| $\lambda$ | Risk aversion |
| $x_t$ | The demand/supply by the agent |
| $E(P)$ | Expected price prediction |
| $P_t$ | Stock price at time t |
| $r$ | Risk-free rate of return |
| $\sigma^2$ | Variance of expected stock price |
| α | Price adjustment constant |
| $d_t$ | Dividend at time t |
| $d_{mean}$ | Dividend mean |
| $\rho$ | Dividend process constant |
| $\varepsilon$ | Dividend process error |
| $P^*$ | Rational expectations equilibrium price |

# 1. INTRODUCTION

## 1.1. THE PROBLEM AND MOTIVATION

The world is facing increasing levels of systems integration, moving towards a complex web of systems that adapt to changing environmental conditions. Business and government applications require integrated systems that exhibit intelligent behavior. The success of complex systems depends on the successful interaction between different groups of smaller systems in order to create a meta-system. Conventionally, the style of operation for businesses and government was to develop or build what they can do and subcontract when they did not have the capabilities. Now, the operation style is to be the lead system integrator where business or government gets the best systems the industry develops and focuses on system engineering, integration, planning and control. This new operation style has led to a new term: System of Systems (SoS). System of Systems describes the interaction between different independent and complex systems in order to achieve a common goal.

Future Combat Systems, NATO, trans-national virtual enterprises, and intelligent transportation systems are some of the networked SoS being observed in governments and commercial enterprises. These networked systems consist of people, organizations, cultures, activities and interrelationships. The semi-autonomous systems (people, organizations) are integrated through cooperative arrangements. It is feasible to understand any System of Systems as a collection of Complex Adaptive Systems. A Complex Adaptive System (CAS) is a collection of independent systems where the emergent behavior of the system is the result of implicit and explicit collaboration of its independent systems. While the individual systems of the SoS can be very different and operate independently, when working together, emergent system level properties can be observed. These systems provide potential for robustness, but also potential for cascading failures. For example, individual electrical utilities form the power grid by connecting electrical utilities from different regions. This formation provides a hidden robustness because each system operates independently and can provide the same capabilities of other linked electrical utility companies. However, at the same time, because of the interdependencies of each component, the SoS is open to cascading failures similar to the

14 August 2003 blackout in Northeast USA. Therefore, stakeholders for SoS must analyze and understand the evolving nature of the emergent patterns before the design and implementation phase to minimize cascading failures. As such, understanding how system behavior emerges from collections of complex adaptive systems is becoming more important. This brings additional challenge to complex system architecting. Architects of SoS need analysis techniques that will not only help in selecting from a large solution space the best architecture that will meet the customer needs, but also help the architects understand the emergent behavior of the architectures.

Traditional analysis techniques have been used to explain the static behavior of the systems. However, these techniques are not efficient enough to explain the adaptive behavioral models of the SoS and CASs under changing objectives. The problem is to develop frameworks appropriate for better understanding of both SoS and CAS. These frameworks should help researchers better understand possible future states of SoS and CAS under different operating or environmental conditions. Besides, humans are an essential component of SoS. Human systems make decisions and facilitate interactions between systems in the SoS. New frameworks should include this component into the analysis. This study is motivated by these challenges for model formulation of SoS that can help system architects understand how SoS evolve and behave in different conditions. Therefore, the focus of this study is on developing frameworks that can capture the emergent behavior of SoS architectures given that humans are also component systems operating under changing environmental conditions.

As an application domain to demonstrate the analysis framework, financial markets are selected since they are a good example of emergence on a grand scale. Financial markets are also a good example of self-organizing systems where there is no centralized control. The financial market regulates prices of companies across the nation, yet there is no entity that controls the workings of the entire market. Investors have limited knowledge of the market and must follow the regulatory rules of the market. Trends and patterns emerge from the transactions of traders. Human systems facilitate market dynamics, so this domain is suitable for incorporating human systems into analysis frameworks for SoS. Markets show rich dynamics, such as volatility clustering, fat-tail distribution, bubbles and crashes, chaos and many more (Chen and Yeh, 2002).

These rich dynamics, along with price formation, emerge from the bottom-up by the behavior of different types of traders, rather than top-down mechanisms. Conventional financial models are not capable of demonstrating these features. In the last decade, there has been increased interest in describing stock markets using computational agent models. This research field, known as the artificial stock market, is distinguished from other traditional methods. Models in this field are composed of heterogeneous interacting adaptive traders. Traditional methods, such as the representative agents, are discarded. The artificial stock market is a good application domain to illustrate SoS modeling since heterogeneous interacting agents represent sub-systems of SoS. At the same time, artificial financial markets are an important artificial intelligence application area for the fields of machine learning since the objectives and interactions of traders tend to be more clearly defined mathematically. The study is also an application area for the SoS problems of distributed intelligence, such as collective learning, coordination and competition. Therefore, the artificial stock market offers a promising approach for studying analysis of different modeling frameworks for SoS.

## 1.2. OBJECTIVES

This research study focuses on developing a framework that can be utilized by system architects to understand the emergent behavior of system architectures. The objective is to design a framework that is modular and flexible in providing different ways of modeling sub-systems of System of Systems. At the same time, the framework should capture the adaptive behavior of the system since evolution is one of the key characteristics of System of Systems. Another objective is to design the framework so that humans can be incorporated into the analysis. The framework should help system architects understand the behavior as well as promoters or inhibitors of change in human systems. Computational intelligence tools have been successfully used in analysis of Complex Adaptive Systems. Since a System of Systems is a collection of Complex Adaptive Systems, a framework utilizing combination of these tools can be developed.

Financial markets are selected to demonstrate the various architectures developed from the analysis framework. This part of the research study focuses on developing artificial financial market with the followed objectives:

- Incorporate long/short position cover mechanisms: Currently, artificial stock market studies model trader behavior through simple buy/sell actions. However, covering to make profit from an earlier investment is an important mechanism in real markets. The effect of this mechanism on market behavior has not been studied through artificial financial markets. By using discrete state transitions for each trader behavior, the effects of these mechanisms on market dynamics are analyzed. This part of the study demonstrates the flexibility of the analysis framework to help system architects understand system behavior under changing rules of engagement or environmental conditions.
- Design intelligent trading agents: The interest is in studying how traders endowed with learning abilities might co-evolve in societies of learning traders. By using techniques from machine learning and artificial intelligence, the effect of adaptive learning on price formation is analyzed. This part of the study demonstrates the adaptation and evolution characteristics of systems.
- Develop behavioral investor model: The behavioral models leading to bias in trader decisions and the effect of various biases to market price formation has not been studied rigorously enough through artificial financial markets. By incorporating Markov based models developed for investor behavior into an artificial financial market, the relation between investor behavior and market dynamics is analyzed. This part of the study demonstrates that the analysis framework can incorporate humans into system analysis and provides means to understand the effect of humans on the emergent behavior of the system architecture.
- Analyze the time series properties of artificial price series: Comparison of statistical properties of the prices generated by the agents to empirically known statistical properties of real markets, such as volatility and the fat-tailed nature of return distributions, are used to validate how much the simulated markets recover known real-world regularities.

The ultimate objective of the application part of this research is not to exactly replicate the financial markets, but to better understand market behavior in a real decentralized financial market. By using simplified models abstracted from the analysis framework for SoS, the aim is to identify real-world regularities in the artificial market and relate how these regularities depend on parameter choices or modeled mechanisms. Finally, these studies will lead to artificial financial market software that can be used as a test tool for better understanding market mechanisms. Better comprehension of the financial market leads to better decision tools and trading strategies. The general framework developed for SoS analysis can be used to analyze the system behavior emerging from different SoS architectures, ultimately leading to better system designs.

## 1.3. APPROACH

In the SoS environment, architecture has more influence on requirements than it does in an environment dominated by one complex system. In a complex system, architecture is the implementation solution for the requirements. However, in a SoS environment the architectural constraints imposed by existing systems can have a major influence on overall capabilities, objectives, requirements and behavior. Therefore, architecture becomes more important in SoS. As a result, this increases the importance of the systems architecting processes. The system architecting process is difficult since there is infinitely large solution space. Several approaches have been used in systems architecting processes to select the architecture that meets customer requirements (Rechtin, 1997). A normative (solution based) technique prescribes a specific architecture for customer needs. However, this approach is not effective in handling major changes in requirements. A rational (method based) technique generates architectures through analytical models using mathematical principles. However, analytical models are not effective in handling the large search space for highly complex systems such as SoS. Participative (stakeholder based) techniques utilize concurrent engineering to minimize the complexities created by multiple stakeholders. This approach focuses on consensus and helps explore the search space, but is nonetheless an undisciplined approach. Heuristics techniques are utilized in system architecting to restrict the search space by eliminating the past mistakes in system design. This approach is useful for minimizing the search space, but it does not provide any insights for system level behavior analysis of architectures.

Simulation modeling has been used as an alternative approach for analyses purposes in systems architecting. This approach provides a description of the system to be built and is specifically useful in systems with a high degree complexity. (Gilbert et al., 1999) identify three periods in the development of simulation: Dynamical systems, micro-simulation and adaptive agent models.

In the 1960's, computer models were used to simulate control and feedback processes in organizations, industries, and cities. These early models consisted of differential equations that described changes in system attributes as a macro-level function of other systematic changes. In the 1970's, simulation models started to use micro-level units for analysis, but retained the emphasis on empirically based macro-level forecasting. In contrast to the macro-level approach in models of dynamical systems, micro simulation is a bottom-up method for modeling the behavior of decision makers within a larger system. This method utilizes representative samples of decision makers, mainly forecasting macro effects that alter individual behavior. Therefore, these models still remain equation-based, much like the earlier dynamical systems model (Macy et al., 2002).

Similar to micro-simulation, the third period in simulation, agent based modeling (ABM), explored the micro-foundations of global patterns. The difference is that agents interact with little or no central authority; they are independent and adaptive and follow simple rules. Traditional methods assume that system or cooperative behavior exists and this upper level produces various forms of social organization and structure. Agent-based modeling assumes that social structure and organization are created from bottom-up via the interactions of individual agents. Rather than examining how social structure shapes behavior, ABM focuses on how local interactions create global social structures. The goal in ABM is to identify the behavioral and environmental mechanisms that create organization and structure in systems.

This research study follows the third wave of simulation and utilizes agent-based modeling for analysis and architecting of SoS and CAS. There is a need for distributed models in representing SoS and CAS. Agent based modeling approach is inherently suitable for this purpose. AnyLogic™ simulation software is chosen as the main tool to build models because the software is a hybrid multi-paradigm simulator capable of modeling systems as a combination of discrete-event, systems dynamics, and agent-based models. This characteristic is especially suitable for simulating complex, dynamic heterogeneous systems. The effect of human systems on overall system behavior is analyzed by incorporating learning classifier systems and Markov-based behavioral processes of traders embedded in an agent-based framework. A learning classifier system is utilized to model learning and adaptation for human systems, whereas Markov-based processes are utilized to model irrational behavior of human traders.

The agent-based approach provides a flexible and modular way of modeling sub-systems of System of Systems and captures the adaptive and emergent behavior of the system architecture. The effect of human systems on the financial market behavior contributes to understanding emergent market dynamics, such as volatility clustering and deviation from efficient market price.

## 1.4. ORGANIZATION OF THE THESIS

The remainder of the dissertation is organized as follows. Section 2 provides definitions for some of the terms that will be used in this study. Section 3 outlines the relationship between System of Systems and Complex Adaptive Systems and provides review of the Artificial Life tools that system architects use in analysis of CAS. Section 4 provides Artificial Life based framework for model formulation of SoS meta-architecture. This section illustrates several different SoS architectures for different systems to explain the framework. The framework is demonstrated with an executable model, the artificial stock market simulation, in Section 5. This chapter includes related literature review, the artificial stock market model, and the initial results. Section 6 presents how human systems can be incorporated into system behavior analysis by outlining the proposed trader architecture, the results and analysis based on the proposed architecture. Finally, Section 7 provides conclusions to the research and summarizes possible future research areas.

# 2. TERMINOLOGY

## 2.1. DEFINITION OF SYSTEM OF SYSTEMS

There are many definitions of System of Systems (SoS) depending on the application area and focus (Maier 2005, Carlock et al. 2001, Sage et al. 2001, Gideon et al. 2005). One general definition of SoS is a mix of multiple systems, which are capable of independent operation interact and collaborate with each other in order to fulfill a global mission. SoS is also a term applied to projects that are large-scale and interdisciplinary with multiple heterogeneous, distributed systems, which are embedded in networks at multiple domains. Several combinations of characteristics are observed in SoS (Bar-Yam et al., 2004):

- Operational independence of elements
- Managerial independence of elements
- Evolutionary development
- Emergent behavior
- Geographical distribution
- Heterogeneity of systems
- System of networks

System of System studies are interdisciplinary and span through the study of architecting, study of various modeling and simulation techniques such as network theory, systems theory, uncertainty modeling, agent-based modeling and object-oriented simulation. The study of numerical and visual tools for capturing system requirements, value engineering, risk analysis, decision and operational analysis are other areas in SoS studies.

## 2.2. DEFINITION OF SYSTEMS ARCHITECTING

System architecting is a process for planning and building of structures and systems to respond to a given need (Rechtin and Maier, 1997). The set of relations, which the architecture describes, can be expressed in various ways such as software, hardware, organizational management or knowledge representation. The essence of system architecting is structuring by bringing form to function, by bringing order out of chaos and converting partially formed ideas of a client into a workable conceptual model. In systems architecting the alternative architectures are large and selection is not easy. Therefore, system architecting process focuses on balancing the customer needs, fitting the interfaces of system components and compromising among the key system attributes, such as cost, risk, schedule and performance.

System architecture is concerned with the internal interfaces among the system's components or sub-systems, and the relationship between the system and its external environment. It is a representation because it provides the elements comprising a system, the relationships among the system elements and the rules governing the relationships. It is also a process because a sequence of steps is necessary to design or change the architecture of a system.

## 2.3. DEFINITION OF AN AGENT

The term Agent describes a software entity that is capable of acting with a certain degree of autonomy in order to accomplish tasks. Different definitions of agents have been proposed by various authors (Russel and Norvig 2003, Nwana 1996, Kaipei et al. 2002). A minimal common definition is given by Feber (1999) as:

An agent is a physical or virtual hardware or software entity:

- which is capable of acting in an environment
- which can communicate directly with other agents
- which is driven by a set of objectives or of a satisfaction/survival function which it tries to optimize
- which possess resources of its own
- which is capable of perceiving its environment to a limited extent
- which has only a partial representation of its environment
- which possesses skills, characteristics and can offer services
- which may be able to reproduce itself

In the Artificial Intelligence field, agents can be comprehended as *intelligent agents* that have the ability to adapt and learn. Intelligent agents are an abstract entity that runs in a dynamic environment and has the adaptation ability to sense the environment and reconfigure in response. This can be achieved through the choice of alternative problem-solving-rules or algorithms, or through the discovery of problem solving strategies. Adaptation may also include other aspects of an agent's internal construction, such as recruiting processor or storage resources (Maes, 1994). Intelligent agents also have the ability to learn through trial-and-error, which results from analysis of behavior and success. Learning can also be through example and generalization, which results from the ability to abstract and generalize (Holland, 1995). When agents are designed to be loosely coupled, it becomes easy to execute them as independent threads on distributed processors. Thus, these agents are called *distributed agents* and the considerations of distributed computing apply.

When several agents interact they may form a *multi-agent system*. Characteristically, such agents will have limited data or methods to achieve an objective and thus will have to collaborate with other agents. Also, in some cases there may be little or no global control and thus such systems are sometimes referred to as *swarm systems*. As with distributed agents, data is decentralized and execution is asynchronous. When agent code starts a copy of itself on another processor and terminates, it effectively moves its execution. This is defined as *mobile agent*.

In terms of agent-based modeling, the basic characteristics of an agent can be given as follows (Kaipei et al., 2002):

- Autonomy: An agent should be an independent and autonomic entity, and it can solve problems independently in random information environment without human intervention.
- Cooperation: An agent should have the ability to interact with other agents and communicate with humans via some communication language.
- Reactivity: An agent should have the ability to perceive and react to the environment.
- Active: An agent should actively take action to other objects.
- Learning: An agent should learn to make itself ingenious when it reacts to or interacts with the external environment.

In different environments, agents have many special characteristics with different tasks. Reaction to environment, autonomy, goal-orientation and persistence are the major characteristics that distinguish agents from other programs. Agents are distinguishable from objects by being more autonomous than objects, by having flexible behavior such as reactive, proactive, and social, and by having at least one thread of control. Expert systems are not agents because these systems are not coupled to their environment, are not designed for reactive or proactive behavior, and are not designed to have social ability (Axelrod, 1997). Agent research programs intersect with complex adaptive system studies, evolutionary game theory studies, multi-agent systems, and micro-simulation studies. The various research areas all try to find answers to common questions, such as:

- How can agents be most effectively combined?
- Which types of hybrid agents will be most relevant to particular problem domains?
- What type of architectures can best integrate the agent's insight mechanisms?

## 2.4. DEFINITION OF ARTIFICIAL LIFE

Artificial life, also known as Alife, is the study of life through the use of analogs of living systems. As defined by Langton (1989), "Artificial life is the study of artificial systems that exhibit behavior characteristic of natural living systems". Christopher Langton founded this discipline in the late 1980s when he held the first "International Conference on the Synthesis and Simulation of Living Systems" (known as Artificial Life I) at the Los Alamos National Laboratory. Artificial life seeks to understand and model systems possessing life that are capable of surviving, adapting and reproducing in sometimes hostile environments (Adami, 1999). It encompasses all the techniques that try to recreate living organisms by computer, including the simulation of behavior processes that result from consciousness and emotions.

Generally, efforts to define life are based on testing for a list of properties. The problem arises from a lack of agreement on what should be included on the list. Properties common to many lists include the ability to replicate, evolve, metabolize, respond to stimuli, and repair damage. Most examples of Artificial Life will fail any such test, unless the list of properties is very short. This life-test list approach is not quite satisfactory. Even if a machine did not replicate, or evolve, or show most of the properties that occur on most life-test lists, it would be hard to deny that it is in some sense alive. Such considerations lead to alternative ways of approaching the problem.

Artificial life researchers have often been divided into two main groups:

- The *strong Alife* group follows Von Neumann's definition of "life is a process which can be abstracted away from any particular medium" (Wolfram 2002; Olson, 1997). Researchers following this belief state that Alife programs are not simulating life in a computer, but are synthesizing it. This approach involves making a long list of properties which are known to occur only in living system, and rather than asking if an example of AL exhibits all items on the list, one asks if it represents an instance of any item on this list. If so, then an instance of some property of life in the synthetic system is captured.
- The *weak Alife* group denies the possibility of generating a "living process" outside of real natural systems. Researchers following this belief try to mimic life processes to understand the appearance of single phenomena. The usual common method is through an agent based model. The researcher is generally interested in some aspect of life, such as evolution, intelligence, language, social behavior, development, etc.

# 3. SYSTEM OF SYSTEMS AS COMPLEX ADAPTIVE SYSTEMS

Considering the characteristics of System of Systems and the characteristics of Complex Adaptive Systems, it is feasible to understand any System of Systems as Complex Adaptive Systems. In order to present the relationship between SoS and CAS, this section provides a review of CAS characteristics as well as Artificial Life tools that system architects utilize in analysis of CAS. This section provides a background for the development of the framework for modeling SoS presented in Section 4.

## 3.1. COMPLEX ADAPTIVE SYSTEMS

Complex adaptive systems are special cases of complex systems. They are made up of multiple interconnected and diverse elements which make them complex. They have the ability to learn from experience which makes them adaptive. The term complex adaptive system was coined at Santa Fe Institute (SFI), by John H. Holland, Murray Gell-Mann and others (Adami, 1999). Holland (1995) defines complex adaptive system as a dynamic network of agents acting and reacting in parallel to what other agents are doing. The overall coherent behavior of the system arises from competition and cooperation among the agents. Figure 3.1 summarizes this definition.

What distinguises CAS from other systems is that agents as well as the system are adaptive. The system is a complex, self-similar collectivity of interacting adaptive agents. CAS's top-level properties are self-similarity, emergence, self-organization and adaptation. Other properties are communication, cooperation, specialization, spatial and temporal organization, and reproduction. All of these properties can be found on all levels. For example, communication and cooperation take place on all levels, from the agent to the system level (Schlagel, 1999). Characteristics of CAS include:

- *Connectivity and interdependence*: Each element in the system is independent and interacts with other elements. The degree of effect of the interaction depends on the connectivity among the elements. Connectivity is not static and changes over time. As a result, connectivity along with interdependence create new order and coherence (Mitleton, 2003). Interactions and the strength of connectivity make it difficult to predict the system behavior (Calvano and John, 2003).

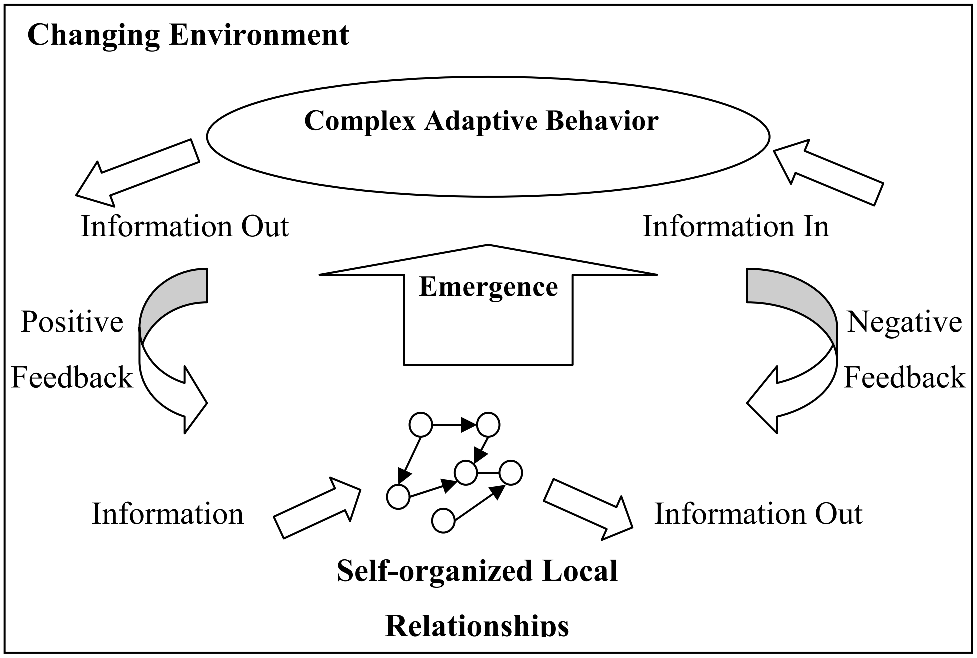


Figure 3.1. Complex Adaptive Systems

- *Co-evolution*: Adaptive moves of each entity alter the landscape of its neighbors (Kauffman, 1995). Therefore, systems evolve with all other related components within the system and other related systems.
- *Dissipative Structures*: Information, energy and matter is exhanged with the environment, pushing structures from equilibrium. As a result, structures vanish, but this also creates new structure and order. The learning capability of complex adaptive systems also indicate that systems have the ability to record history (Mitleton, 2003).
- *Exploration of the space of possibilities*: An optimal solution for a CAS does not exist because CAS is situated in a changing environment. An optimum solution for one specific environment can be the worst solution when the environment conditions change. Therefore, CAS should try to find different strategies (Mitleton, 2003).
- *Feedback and path dependence*: Since CAS is sensitive to impacts from environment, small causes may necessitate re-architecture of the system.

> Feedback is the most important part of the re-architecting process. The initial conditions and the system's past history have an effect on the specific paths the system may follow (Mitleton, 2003).

The top-level properties of CAS, emergence and self-organization, have drawn special attention. The following two sub-sections focus on these two properties.

**3.1.1. Emergence.** Emergent behavior arises from the interactions of elements and the outcome cannot be predicted from studying only the elements of the system or the knowledge of the original conditions. The natural evolution of the complex system can yield unpredictable results, called emergent properties (Kilicay N. and Dagli C., 2003a, b). The property itself is often unpredictable and unprecedented, and may represent a new level of the system's evolution (Ronald et al., 1999).

There are two major reasons why emergent behaviour occurs: complex relations across different levels of the system and feedback mechanisms (Ronald et al., 1999). The components of the system increases combinatorially, which can result in new types of behaviour to emerge. However, having a large number of interactions is not enough by itself to guarantee emergent behaviour since many of the interactions may be negligible or irrelevant, or may cancel each other out. In some situations, a large number of interactions can create noise and can work against formation of emergence. Therefore, it is not only the number of connections between components that determine emergence but also how the connections are organized (Kubik, 2003). In some cases, the system has to reach a combined threshold of diversity, organisation, and connectivity before emergent behaviour appears (Kubik, 2003). There is no scientific consensus about weak and strong forms of emergence and how emergence can be identified (Kubik, 2003), or how much emergence should be used as an explanation in general.

**3.1.2. Self-organization.** Self- organizations involves the coming together of parts spontaneously and endogenously to perform one objective. It is a set of dynamical mechanisms in a system where structures that are not externally imposed appear at the system level. For a system to show self-organized characteristics, the following properties should exist (Nicolis and Prigogine, 1987):

- Multiple interactions with nonlinear dynamics
- Balance of exploration and exploitation mechanisms

- Positive feedback, such as reinforcement, recruitment
- Negative feedback, such as competition, exhaustion

The internal mechanisms are random and amplify random fluctuations that break symmetries. Random walks, errors, random task-switching are some of the internal mechanisms that result in self-organized properties. Also dissipative structures of the system affect the environment, resulting in a continuous change. Besides these properties of the system that may lead to self-organization, the following indicators of self-organization should also be considered (Bonabeau et al., 1999):

- Dissipative structures arise in an initially homogeneous medium.
- Several stable states exist in the system-level behavior and the one that is actually reached depends on the initial conditions.
- Slight variations of some system parameters may lead to dramatic changes in system behavior (bifurcations).

Self-organization is sometimes combined with emergence. However, self-organization can occur without emergence and emergence can arise without self-organization. The link between emergence and self-organization is another research area that remains active (Lansig, 2002).

### 3.2. SYSTEM OF SYSTEMS VS COMPLEX ADAPTIVE SYSTEMS

The relation of SoS characteristics and CAS characteristics are outlined in (Correa and Keating, 2003). Table 3.1 summarizes the relationship between CAS and SoS.

In terms of non-linearity, interdependence and evolution, SoS share similar characteristics with CAS. In terms of self-organization, SoS can be designed with full control over sub-systems. However, full control is not possible for most SoS, so sub-systems self-organize to achieve a goal. For self-organizing systems, control can be achieved through changing environmental or operational rules. From the analysis of the properties of SoS and CAS, it is reasonable to conclude that SoS consists of one or more CAS.Therefore, this study will analyze SoS as collections of CAS and will utilize the CAS analysis tools to derive a framework for model formulation of SoS. The rest of this section reviews some of the CAS analysis tools, specifically computational intelligence tools in Artificial Life studies.

Table 3.1. Comparison of Properties of SoS and CAS

| Characteristics | CAS | SoS |
| --- | --- | --- |
| **Non-linearity** | System level behavior can not be deducted from the behavior of lower level components of the system. | A meta-system behavior cannot be derived by analyzing the behavior of the component systems. |
| **Interdependence** | Each element in the system is independent and interacts with other elements of the system | A meta-system is created by connecting independent systems together. |
| **Evolution** | Feedback from the environment leads to adaptation in system elements. This leads to re-architecting in the system. | As the requirements or environmental conditions change, the meta-architecture evolves. This requires adaptation in sub-systems. |
| **Self-organization** | Feedback, adaptation and non-linear dynamics lead to elements organizing without any control. These systems can only be controlled through changing the rules of engagement of the environment or the system. | There are SoSs where there is full control over the sub-systems. There are also SoSs where full control is not possible and sub-systems self-organize to achieve a goal. |

## 3.3. COMPLEXITY THEORY

Complexity Theory can be defined as a science of complexly interacting systems; it explores the nature of interaction and adaptation in such systems and how they influence such things as emergence, innovation, and fitness (Bar-Yam, 2003). Most attention is given to the complex adaptive systems: how they work, their behavioral model, the reasons of complexity. Complexity Theory is used as a broad term for addressing the study of complex systems, including studies such as systems dynamics, social dynamics, chaos theory, and artificial life.

System dynamics is a method for understanding the dynamic behavior of complex systems. This method focuses on the structure of the system that is the relationships among its components. System dynamics depends on the concept that non-linear feedback can create a vast complexity of emergent behavior from simple activities. System dynamics does not focus on prediction like traditional linear modeling techniques; it emphasizes capturing an understanding of the dynamics of the system (Matthews and Collier, 2000).

Chaos theory deals with certain nonlinear dynamic systems, which under certain conditions exhibit chaotic behavior. These systems have sensitivity to initial conditions known as the "butterfly effect" - a small change in the initial condition of the system causes a chain of events leading to large-scale phenomena. Social dynamics is a mathematically inspired method to analyze societies based on systems theory and sociology. It focuses on the ability of the society to react to inner and outer changes and deals with regulation mechanisms (Axtell, 2003).

Complexity Theory is a beneficial approach to define and understand the concept of identity of a system. It helps in understanding how complex systems are affected from their environments and how a system learns by proposing alternative ways for improvement. It also answers the question of why some good predictions and solutions can be obstructed by dynamic nature of the environment. Finally, it provides an understanding that considering the interactions which shape the system's future behavior is a much more effective endeavor than trying to predict outcomes of the systems. There are some conclusions that Complexity Theory arrives at (Levy, 2000):

1. *Long term planning is impossible:* There are non-linear relationships among components of complex systems, therefore, behavior of complex systems appear random. Therefore, long-term planning is impossible. System of Systems is composed of complex adaptive systems thus the same property applies for SoS.

2. *Dramatic change can occur unexpectedly:* Traditional studies about systems assert that each effect creates its reaction and small effects can also cause small changes in the

nature of complex systems. However, Complexity Theory reconsiders this conclusion and claims that small perturbations can also cause huge changes on the overall system behavior. This property is the reason for cascading failures in SoS.

3. *Complex systems exhibit patterns and short-term predictability*: Instead of searching order in complex systems, randomness of complex systems should be studied. Long-term forecasting is impossible, but short-term forecasting and describing the behavioral model of systems is possible. Therefore, next time period behavior of systems can be predicted when reasonable specifications of conditions at one time period are given. System of Systems testing and validation is based on this characteristic.

4. *Organizations can be designed to be more innovative and adaptive:* Complexity Theory suggests that emergent order and self-organization provide a robust solution for organic networks to be successful in competitive and rapidly changing environmental conditions. System architects can benefit from this property of complex systems by designing SoS components that can self adapt and self organize to changing environmental and operating conditions.

## 3.4. SYSTEM ARCHITECT'S TOOLBOX: THE ROLE OF ARTIFICIAL LIFE

In this sub-section, some of the basic concepts and technologies of research into complex adaptive systems are presented. The review presents approaches developed in biology, physics, and in different branches of computer science. Equilibrium statistical physics, population biology and ecological modeling are branches in physics and biology that deal with complex systems. State models for performance assessment, block diagrams, rule-based models are other tools that are used for dealing with complex systems. The purpose of this review is not to cover all research activities and all the specific results, but rather to gain a basic understanding of the characteristics of such systems by looking at the same class of systems from different Artificial Life study perspectives.

The history of Artificial Life foundations go back far to Neumann's work on cellular automata, Grey Walter's work on reactive robots, and Warren McCulloch's work on the creation of neurons (McCulloch, 1965). More recently, the issues of Artificial Life were introduced by C. Langton as "the study of life as it might be and not of life as it is"

(Langton, 1989). Therefore, the aim of Artificial Life studies is to abstract the underlying principles of the organization of living things and implement them in a computer so that complex system can be studied and tested for various controlled conditions. Artificial life is a meeting point for many disciplines, including traditional fields such as linguistics, physics, mathematics, philosophy, computer science, biology, anthropology and sociology in which unusual computational and theoretical approaches (that would be controversial within their home discipline) can be discussed.

In engineering science there is an expectation that the natural events in the real world can help to predict implications and behaviors of complex systems. The methods and the results of research in other sciences, such as biology, ecology, economy, and computer science can be adapted to the study of complex adaptive systems. The most known natural event analysis study is the Reynold's flock of bird simulation in the field of ecology (Macy and Willer, 2002). This study not only provides better understanding of the dynamics of flock of bird movement, but also has lead to analysis of other colony-based animals such as ant colonies, swarm of bees and termites. These studies inspired the development of swarm intelligence algorithms that are used in analysis of complex systems. Another study done by Hall (1998) emphasizes employing fractal geometry and its non-linear dynamics to the study of complex systems. In his study, a fern is used and he observes that each shape in ferns is repeated in several scales. The chaotic nature of leaf of a fern grows to reach every cell in the leaf. This study illustrates that simple non-linear function can create incredible complex behavior when iterated. The fractal units are useful tools to solve the problem about coordination of elements in systems. Calvano and John (2004) examine the applicability of power law relationships to complex adaptive systems. They conduct their research to answer the question that conclusion from observed natural systems can be extended to define the behavior of complex engineering systems. Aside from the studies of natural events as tools to understand behaviors of complex adaptive systems, the field of Artificial Life now extends over several main research topics, including the following:

- Analysis of complex phenomena with the aid of cellular automata or non-linear differential equations.
- Evolution of populations through the use of evolutionary algorithms.

- Creation of animats (animal robots); autonomous creatures capable of acting and surviving in a changing environment.
- Study of collective phenomena based on the interaction of an assembly of reactive agents.

Following sub-sections provide some information about specific fields of Artificial Life.

**3.4.1. Cellular Automata.** A cellular automaton is a collection of “colored” cells on a grid of specialized shape that evolves through a number of discrete time steps according to a set of rules based on the states of neighboring cells. The rules are then applied iteratively for as many steps as desired (Wolfram, 2002). Cellular automata come in a variety of shapes.

The three fundamental properties of a cellular automaton include: The type of grid on which it is computed, the numbers of colors the cells assume, and the neighbors over which cells affect each other. The simplest grid is a one-dimensional one. Two dimensions such as square, triangular and hexagonal grids can be considered. The best-known cellular automaton is Conway's game of life (Gardner, 1970). This is a two dimensional grid where the rules include: If a black cell has 2 or 3 black neighbors, it stays black. If a white cell has 3 black neighbors, it becomes black. In all other cases, the cell stays or becomes white. Despite its simplicity, the system achieves an impressive diversity of behavior, fluctuating between apparent randomness and order (Wolfram, 2002). Cellular automata models are applicable to a wide range of research topics. They are used in the study of various aspects of the world, including manufacturing, communication, computation, construction, growth, reproduction, competition, and evolution (Margolus and Toffoli, 1987) (Crutchfield et al., 2003).

**3.4.2. Agent-Based Models.** Agent based modeling is a computational method where a system is modeled as a collection of autonomous decision-making entities that interact in non-trivial ways. It consists of a set of agents and framework for simulating their decisions and interactions. ABM is related to a variety of other simulation techniques, including the discrete event simulation and distributed artificial intelligence or multi-agent systems. Although many traits are shared, ABM is differentiated from these approaches by its focus on achieving simplicity (Axelrod et al., 1996). In other words, although agent-based modeling employs simulation, it does not aim to provide an accurate representation of a particular empirical application. The goal of agent-based modeling is to enrich our understanding of fundamental processes that may appear in a variety of applications.

Agent-based modeling is a viable way to study agents who are adaptive rather than fully rational. This is necessary because the interactions of adaptive agents typically lead to nonlinear effects that are not suitable to the deductive tools of formal mathematics. Axelrod (1997b) defines ABM as a third way of doing science because it is similar to deductive and inductive methods in some ways, but it is also different from deductive and inductive methods. ABM starts with a set of explicit assumptions similar to deductive methods, but unlike deduction it does not try to prove any theorem. ABM is similar to inductive methods because it generates data that can be analyzed inductively. However, unlike inductive methods, the simulated data are generated from a specified set of rules rather than direct measurement of the world (Axelrod, 1997b).

ABM is widely used in many applications including manufacturing, control systems, automated systems, financial market analysis, social sciences and even anthropology. Some of the early influential studies are reviewed in this section. Financial market studies deploying ABM are reviewed in section 4 separately. One of the earliest and most famous studies is Schelling's residential tipping simulation (Macy and Willer, 2002). It provides a good example of a simple model that provides an important insight into a general process. The model assumes that a family will move only if more than one third of its immediate neighbors are of a different type (race or ethnicity). The result is that very segregated neighborhoods form, even though everyone is initially placed at random and everyone is somewhat tolerant.

Epstein and Axtell (1996) construct an artificial society where agents live in a two dimensional square grid containing renewable resource of sugar. Every agent is born into this world with a metabolism demanding sugar, and each has a number of other attributes, such as visual range for food detection, that vary across the population. They move from square to square according to a simple rule: Look around as far as your vision permits, find the unoccupied spot with the most sugar, go there, and eat the sugar. At its simplest level, the Sugarscape model represents a kind of hunter-gatherer society. The model reproduces the kind of strongly skewed distribution of wealth generally observed in human societies - where a few individuals hold most of the wealth and the bulk of the population lives in relative poverty. The Sugarscape model also offers insights into other phenomena, such as the introduction of trade (Epstein and Axtell, 1996).

Bunn and Oliveria (2001) construct an agent based computational model of a wholesale electricity market to explore the possible effects of the New Electricity Trading Arrangements (NETA) introduced in the U.K. in March 2001. Tassier et al. (2002) implement an agent-based model to study a range of consumer behaviors in a monopolistic durable goods market using the automobile industry as an example. They use agent-based modeling as an extension of standard theory, thus demonstrating a complementary between standard models and agent-based models of economic theory.

Norms provide a powerful mechanism for regulating conflict in groups, even when there are more than two people and no central authority. A norm exists in a given social setting to the extent that individuals usually act in a certain way and are often punished when seen not to be acting in this way. One of the most influential studies about norms was done by Epstein (2001), where he investigates the emergence and stability of behavioral norms in the context of a game played by people of limited rationality. Agent-based simulations of the norms game and meta-norms game have allowed the exploration of important dynamics of norms. It shows that relying on individuals to punish defections may not be enough to maintain a norm. Other mechanisms should be established to support norms. Results show conditions under which norms can evolve and prove stable.

**3.4.3. Evolutionary Algorithm.** Evolution is another area where biological analogies are used in system design. Evolutionary principle can be thought as the consequence of any one of three different mechanisms (Axelrod, 1997a). First, it could be that the more effective individuals are more likely to survive and reproduce. The second interpretation is that players learn by trial-and-error, keeping effective strategies and altering ones that turn out poorly. The third interpretation is that players observe each other and those with poor performance tend to imitate the strategies of those they see doing better.

**3.4.3.1 Genetic algorithms.** Genetic Algorithms (GA) are utilized for simulating evolutionary processes including economic learning (Holland J. 1975, Goldberg D. 1989, Mitchell M. 1996). Lettau (1997) outlines the advantages of Genetic Algorithms. GAs cover different regions of the search space. Too much exploitation can result in missing the global optimum, but too much exploring hurts system performance. Therefore, the balance between exploration and exploitation is an important component of adapting and learning systems. GAs balances the trade-off between exploration and exploitation.

Axelrod's study of the prisoner's dilemma is one of the best known studies that focus on cooperation. Axelrod (1984) conducted computer simulations to find out which strategies worked best for prisoner's dilemma. For the simulations, he used genetic algorithms and simulated using a population of twenty individuals per generation. Results show that genetic algorithm evolved populations whose members were as successful as the tit-for-tat strategy, which involves cooperating on the first move and then doing whatever the other player did on preceding move. Some rules evolved to be more effective than tit-for-tat strategy and broke. These rules are better in particular environments and not robust in other environments. Other influential GA applications are reviewed in (Mitchell and Forrest, 1998).

**3.4.3.2 Learning classifier systems.** The learning classifier system has three main components: the performance component, the reinforcement component, and the discovery component (Wilson, 1995). Signals from the environment are received by the classifier system and several rules whose conditions are satisfied compete for final execution. At the performance component, rules enter competition according to their strength (fitness value). Once the decision is selected and profit is known, the reinforcement component rewards the rules which predicted the outcome correctly by increasing their fitness values and punishes rules that are not correct. At the discovery component, genetic algorithms are used to evolve better rules. The GA is applied at random times to each agent and replaces rules by new ones using crossover and mutation.

Learning classifier systems are used in financial market studies to model the learning mechanisms of traders (Palmer et al., 1994) (LeBaron, 2001, 2002). They are also used to model the behavior for animats (Dumeur R., 1991), discovery of novel maneuvers in simulated combat (Smith et al., 1999) and to model control mechanisms of unmanned vehicles in unknown environments (Cazangi et al., 2003).

**3.4.3.3 Genetic programming.** Genetic programming is the extension of the genetic model of learning into the space of programs. The objects that constitute the population are not fixed-length character strings that encode possible solutions to the problem at hand, they are programs that, when executed, are the candidate solutions to the problem (Koza, 1992). These programs are expressed in genetic programming as parse trees, rather than as lines of code. Genetic programming is a machine learning model which is general and flexible and has already been applied to a wide variety of problem domains, including financial markets (Chen, 2003).

**3.4.4. Animats.** The hypothesis of animat approach is that by simulating animal-like systems at a simple level, humans can be simulated gradually. Full connection with a sensory environment with maximum use of perception and adaptation is included in animats so that when human level is reached, these elements will be available. This study area also hopes to reach human intelligence from bottom-up instead of high level competences. Survival needs are the principal drivers of animal behavior. The effect is that survival needs have influence on formation of inductive bias and animat approach explicitly makes them drivers of system behavior (Wilson, 1991).

The basic strategy of animat approach is to work thorugh higher levels of intelligence from below using minimal ad hoc machinery (Wilson, 1991). The process is incremental: Given an environment and an animat with needs, a sensory/motor (architecture) system satisfies these needs. By increasing the difficulty of the environement or the complexity of the needs, the minimum increase in animat complexity necessary to satisfy the needs are searched. Alternatively, the environment can stay the same but the needs satisfaction criterion can be increased. Similarly, the mininal animat complexity increase is searched.

**3.4.5. Cognitive Architecture Studies.** Cognitive architecture studies specify the underlying infrastructure for an intelligent system. These studies describe the system in two components; the architecture and knowledge. The architecture is composed of mechanisms that are fixed and reusable across applications. Since most problems are not purely rational or purely reactive, hybrid cognitive architectures in the form of layers are presented (Flores-Mendez, 1999). These architectures have several layers to deal with different level of abstractions. Soar and ACT-R are two hybrid cognitive architectures that support most of the cognitive mechanisms. Soar is developed from an artificial intelligence viewpoint; ACT-R is developed from an experimental psychology viewpoint.

ACT-R is composed of sensory modules, action modules, and intentional module for goals. Each module has buffers for short-term memory. The long-term production memory coordinates all the modules in ACR-R (Langley et al., 2006). Soar also has a long-term memory consisting of production rules. It also has a semantic memory for holding previous states and an episodic memory holding previously seen facts. Soar has several learning mechanisms, such as chunking, reinforcement learning, and semantic and episodic learning (Jones, 2004).

Sloman's (2002) H-Cogaff cognitive architecture is another hybrid human-like information processing architecture. The H-Cogaff architecture meets the requirements of a complex adaptive system analysis because it represents a combination of the cognitive architecture and the MAS conceptual frameworks (Taylor et al., 2005). It provides a framework for describing different kinds of architectures and sub-architectures. It consists of perception, central-processing and action components. The central-processing component has three-tiered sub-architectures, which are reactive, deliberative and meta-management mechanisms. The reactive layer responds immediately to sensor input, whereas middle layer components enable decision making, planning and deliberative reasoning. The third layer supports monitoring, evaluation, and control of internal process in the lower layers. Figure 3.2 summarizes this architecture.

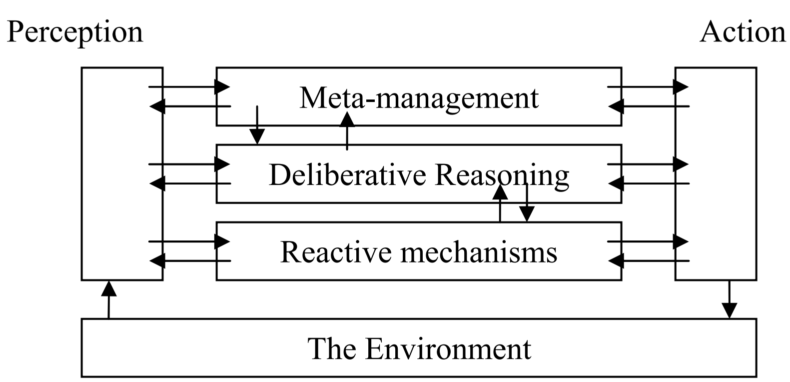


Figure 3.2. H-Cogaff Architecture

**3.4.6. Swarm Intelligence.** Swarm intelligence specifically focuses on collective intelligence. The characteristic of collective intelligence is that many agents run concurrently performing actions which affect the behavior of other agents. Centralized and personalized communication is not allowed. Also there has to be a well-specified task set for the entire distributed system that requires maximizing some utility function.

Stigmergy is the emergence of coordinated system-level behavior from local interactions of individuals (Bonabeau, Dorigo, Theraulaz, 1999). Simple activities may be coordinated by indirect communication and robust phenomena may emerge that remain virtually unchanged even under changing circumstances. Two general forms of stigmergy are possible: One form involves a change in the physical characteristics of the environment. An individual observes a developing structure and adds to it (like termite nest building). The other form is sign-based stigmergy. Some marker is deposited in the environment that makes no direct contribution to the task being fulfilled but influences the subsequent task related behavior. Stigmergy does not explain the detailed coordination mechanisms. For designing a system to fulfill a task, it provides a general concept that links individual and colony level behavior. This mechanism allows simple agent construction, reduced communications and flexibility, and robustness of the system level behavior in the face of disturbances (Bonabeau, Dorigo, Theraulaz, 1999).

**3.4.6.1 Ant colony optimization.** Ant colony optimization (ACO) is a specific application of the swarm intelligence approach that seeks to adapt coordination mechanisms employed in social ant colonies to solve discrete optimization problems (Bonabeau et al., 2001). ACO artificial ants build solutions by moving on the problem graph and, by mimicking real ants, deposit artificial pheromone on the graph in such a way that future artificial ants can build better solutions. ACO has been successfully applied to an impressive number of optimization problems, such as the traveling salesman problem, to dynamic real-world problems like routing and load-balancing in circuit switched telecommunications networks (Bonabeau et al., 2001).

**3.4.6.2 Particle swarm optimization.** Particle swarm optimization (PSO) is a global minimization technique for dealing with problems in which a best solution can be represented as a point or surface in an n-dimensional space. The technique is developed by (Eberhart and Kennedy, 1995), inspired by social behavior of bird flocking or fish schooling. In flocks of birds or swarm of fish, if one sees a desirable path to go (for food, protection, etc.) the rest of the swarm will be able to follow quickly even if they are on the opposite side of the swarm. This is modeled by particles in multidimensional space that have a position and a velocity. The particles fly through hyperspace and remember the best position that they have seen. Members of a swarm communicate good positions to each other and adjust their own position and velocity based on these good positions. There are two main ways this communication is done: a swarm best is known to all, while local bests are known in neighborhoods of particles. PSO has been applied to replace the back-propagation learning algorithm used with artificial neural networks (ANN). It is faster and gets better results in most cases. It also avoids some of the problems of GAs (Engelbrecht, 2002). PSO is also applied to swarm robotics (Mondada et al., 2003).

**3.4.7. Multi-Agent Systems (MAS).** In contrast to agent-based modeling where the focus is on analysis of emerging system behavior, multi-agent studies focus on design aspects of the agents, such as their interaction and communication structures under various environmental conditions (Kilicay et al., 2006a). Some of the major study areas can be described as follows (Flores-Mendez, 1999):

- Agent Architectures
- Agent-System Architectures
- Agent Infrastructures

**3.4.7.1 Agent architectures.** Agent architecture studies focus on the internal architecture of agents, such as such as perception, reasoning, and action components. Since multi-agent systems are constructed without any global control, one way to prevent chaotic behavior of the system is to design perception and reasoning into agents. Each agent can form expectation models of behavior for other agents, and can reason about global effects of local actions that ultimately can lead to coherence in the system (Sycara, 1998). Two dominant agent architectures differ conceptually by the way they look at intelligence. The Belief-Desire-Intension (BDI) agent architecture designs agents assuming intelligence emerges from rational behavior, whereas Reactive agent architecture designs agents assuming intelligence emerges from simpler behaviors of interaction between an agent and its environment. BDI types of agents have sophisticated reasoning mechanisms that integrate planning, scheduling, information gathering, and coordination with other agents (Ferber, 1999). As opposed to BDI agents, reactive agents do not take history in account or plan for the future. Instead, they respond to the present state of the environment. Various forms of reactive agent architectures are reviewed in (Ferber, 1999). Since most problems are not purely rational or purely reactive, hybrid architectures in the form of layers are presented (Ferber, 1999). These architectures have several layers to deal with different level of abstractions. Usually the lowest level makes decisions based on raw data, a middle layer creates a knowledge-level view of the agent's environment, and the upper level deals with the interaction of an agent with its environment and other agents. The interaction between these basic layers can be various resulting in different architectures, and ultimately different system behaviors (Sycara, 1998).

**3.4.7.2 Agent-system architectures.** Agent-system architectures analyze agent interactions and organizational architectures where agents operate and interact under specified environmental constraints. One way of forming system architectures is based on structure of information and control relations between agents. Another path in forming system architectures is based on organization theory where sets of agents with mutual goals, characteristics, or beliefs are organized into groups. This type of organizational structure forces different coordination and communication structures among agents. Hierarchical organization (where superior-subordinate relationship exits), specialist agents organization (where each group is an expert in one part of the problem), and market organizations (where agents interact only through price variable) are some example system architectures developed in MAS research (Sycara, 1998).

**3.4.7.3 Agent infrastructures.** Agent infrastructure studies focus on interface mechanisms of multi-agent systems, which mainly involves communication aspects between agents. These studies try to achieve a common agent communication language and protocols, common format for the content of communication, and shared ontology between agents. One of the popular agent languages is KQML (Knowledge Query and Manipulation Language), which consists of three layers - communication layer, message layer and content layer (Flores-Mendez, 1999).

**3.4.7.4 Design methodology.** Apart from the architectural design studies on components of the multi-agent systems, several conceptual frameworks are proposed for design of multi-agent systems. The objective in these studies is to provide a general systematic methodology for designing agent-based systems. Wooldridge's (2000) framework deals with macro- and micro-level aspects of the design and is neutral to the application domain and agent architectures. The framework considers agent-based systems as artificial societies and defines the system in terms of roles, which are further defined in terms of responsibilities, permissions, protocols, and interactions. Each attribute is modeled in detail in analysis and design phases of the framework. This framework is suitable for small size multi-agent systems of less than 100 agents.

Burmeister's framework (1996) is an extension of object-oriented techniques and defines three basic models - agent model, organizational model, and cooperation model. The agent model defines the internal agent structure, the organizational model outlines relationships between agents, and the cooperation model describes interactions between agents. The agent oriented methodology for enterprise modeling framework (Iglesias, 1998) is geared towards manufacturing applications. Object-oriented, enterprise modeling and computer integrated manufacturing open system architecture methodologies are combined into one framework. Iglesias et al. (1998) provide a detailed survey of agent-oriented methodologies. Among these methodologies are both extensions of object-oriented strategies, as well as extensions of knowledge engineering methodologies.

## 3.5. SUMMARY

Figure 3.3 summaries the currently available computational intelligence tools that are utilized in analysis of Complex Adaptive Systems. Methods such as evolution, swarm intelligence, agent-based modeling, and synthetic ecosystems focus on system behavior, whereas distributed artificial intelligence and multi-agent systems focus on system design and architectures. At the intersection of all these methods is Artificial Life, which combines both views of system behavior and system design by utilizing any combination of these tools.

Even though many diverse analysis perspectives exist for analysis of SoS, no formal methodology has been yet developed (Correa and Keating, 2003). The diversity,

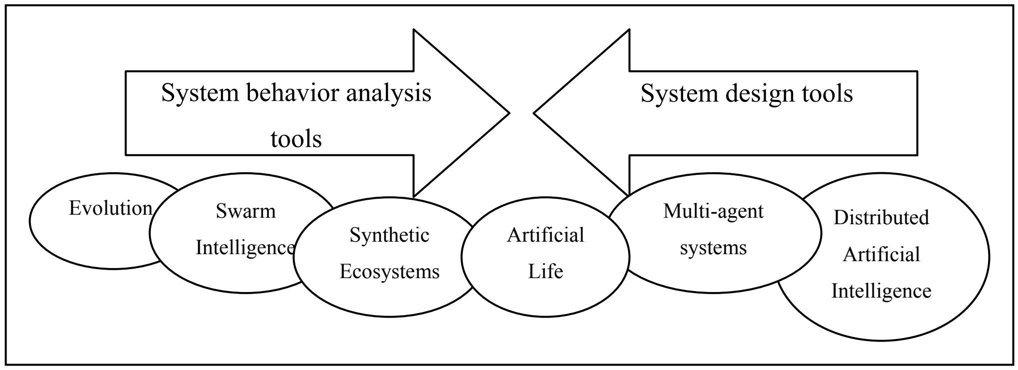


Figure 3.3. Emergence of Artificial Life Research

complexity and scale of these systems require integration of multiple methodologies. Considering SoS as collections of CAS, the Artificial Life methodology captures system from both the architecture and behavior perspectives. Therefore, in the next section Artificial Life based framework for SoS modeling and analysis will be described.

# 4. ARTIFICIAL LIFE FRAMEWORK FOR MODEL FORMULATION OF SYSTEM OF SYSTEMS

System of Systems conceptual framework identifies three components: *physical networks* such as roads and power grids, *information networks* such as databases, Intranets and *social networks* such as people, organizations and processes. One of the desired goals of SoS architecting is to create robust physical networks, information networks and social networks and integrate these three main components seamlessly. This can be achieved through better networking, which can lead to improved situation awareness, which enhances collaborations and interactions in social networks leading towards more effective SoS. Even though the conceptual framework outlines the steps to successful SoS architecting, there are many challenges.

Continuous rapid technological changes provide opportunities for improved capabilities, but increase complexity of interfaces as well as interoperability between legacy systems and new systems. Dynamically changing requirements increase uncertainty in architecting processes. The need to design dynamic architectures, which achieve a diverse spectrum of missions and operations, is another major challenge. All these challenges open various research needs for SoS. Maier (2005) identifies several research areas specific to SoS. One area is to balance the socio-technical equilibrium of SoS. This becomes important in social SoS such as intelligent transportation systems. Designers are challenged with explicitly incorporating interactive social and technical effects into system design. This enhances the need for incorporation of human systems into models of SoS. Another challenging research area is the adjustment of optimization techniques for identifying invariant architectures that will be useful for many design solutions rather than an optimal solution to a specific problem. Another research area is the need for better upper level descriptive and analysis frameworks for SoS. For example, state models and simulation of state models are used in performance assessments of Future Combat Systems (Campbell et al., 2005). DoDAF Architecture, ISO Reference Model for Distributed Processing, and SysML are upper level descriptive frameworks used for SoS analysis. Apart from these descriptive frameworks, there is a need for frameworks that can capture emergent behavior of system architectures. This section describes Artificial Life based framework for analysis of emergent behavior of system architectures. The following sub-section describes the SoS as meta-architectures, while the rest of the section discusses how the framework formulates models for the SoS meta-architectures.

### 4.1. COMPLEX SYSTEMS ARCHITECTING

Complex systems architecting is an attempt to integrate several complex systems into meta-architectures. From many potential component systems, a set must be selected to construct the meta-architecture for SoS. The selection of the set depends on the requirements, functionalities and capabilities desired from the SoS to achieve the common mission. Since the meta-architecture operates in continuously changing environments, multiple system states and actions must be explored during complex system architecting processes. Also, spiral development of SoS necessitates dynamically changing evolving architectures (Kilicay et al., 2007a). This requires the creation of a meta-architecture that consists of core components that remain unchanged for a given period as other components are evolved in time.

To achieve architecting such a meta-system, all component systems need a physical global interface to function. Initially, the Department of Defense defined the Global Information Grid (GIG) as the seamless communications architecture for information superiority and the basic interface for creating meta-architectures for United States Military (Buda et al., 2001). Now, the Global Information Grid represents the system formed by the distributed collections of electronic capabilities that are managed and coordinated to support some sort of enterprise (virtual organization). Different independent systems are connected to the GIG to create a network-centric architecture. Therefore, an evolving physical architecture is created by connecting systems to GIG. This net-centric architecture is also evolving to meet the changes in system requirements and objectives. It is the dynamically changing architecture that creates the best net-centric systems, although data is a necessity for the system to function. A dynamically changing meta-architecture for System of Systems can be defined as a collection of different Complex Adaptive Systems that are readily available to be plugged into the evolvable net-centric architecture. Figure 4.1 illustrates this meta-architecture (Dagli et al., 2007).

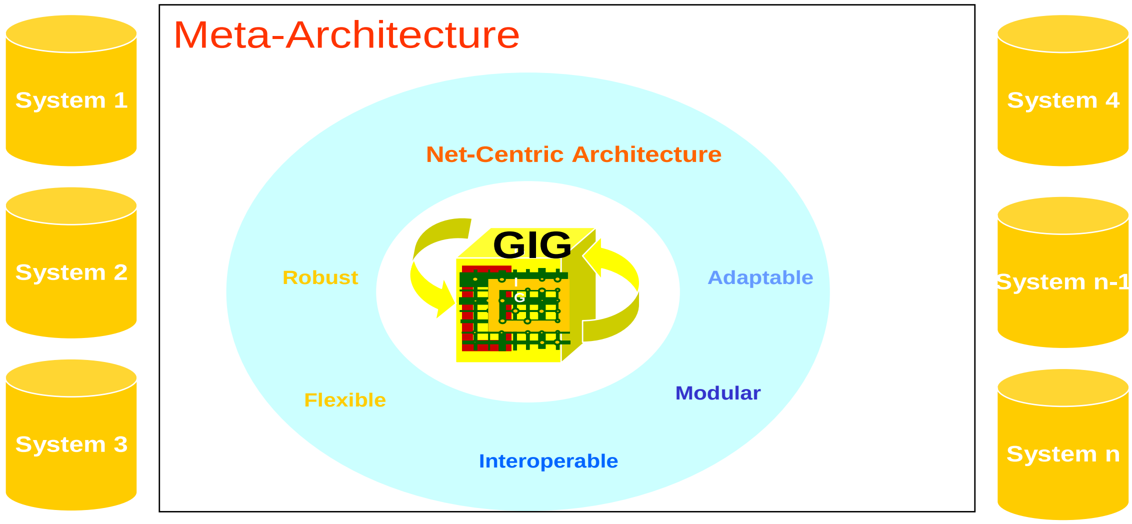


Figure 4.1. Meta-architecture Generation

Modeling of SoS requires a procedure similar to system architecting. Scoping the model, selecting the model attributes, partitioning the model into sub-components, and then aggregating sub-components into one system that can represent the system requires modular and flexible modeling framework. Section 4.2 discusses a modular framework for modeling the SoS meta-architecture.

## 4.2. THE FRAMEWORK

System of Systems comprises social, physical and information domains. Frameworks for modeling SoS should focus on integrating these three different views into one seamless model. These models should also incorporate humans as component systems into a SoS model. This becomes especially important at the refinement and exploration phase of system architecting. Since humans operate as component systems in SoS, the frameworks for modeling these systems should also incorporate human behavior into systems analysis. This can provide insights about system behavior under different social behaviors at the architecting phase.

Cognitive architectures have been used on the front-end analysis portion of systems engineering (Madni et al., 2005). Cognitive architectures represent a promising approach to explaining mental processes and human behavior with error generation mechanisms. Cognitive architectures embedded within system architectures are useful in identifying the effect of human errors on the overall system behavior. On the other hand, multi-agent models are a suitable tool for modeling SoS because they provide means of integration for the social, information and physical components of SoS. Figure 4.2 provides the framework for modeling CAS and SoS. It consists of several layers for modeling different components of SoS. Layering the framework is important for keeping the architecture simple at each layer. Systems at low layers become simple components at the higher level and aggregate components disappear at the highest level. Therefore, the framework consists of several layers: computational intelligence tools, mechanism modules, cognitive architecture, agent level, environment level and system level.

**4.2.1. Sub-system Models.** The computational intelligence toolbox, which contains the tools discussed in Section 3, is used to design the mechanism modules that are sub-components of cognitive architectures. One or more combination of these modules shapes the agent architecture of the system.

At the cognitive level, Sloman's H-Cogaff architecture is selected because this architecture is modular and flexible to model different sub-components of SoS. Besides, different architectures can be compared and contrasted using this general representation. For example, in some designs, deliberative reasoning layer dominate the cognitive architecture, but in some designs high levels lose control to reactive layers.

The cognitive architecture embedded in the multi-agent model provides different ways of modeling sub-systems or sub-components. At the agent level, agents can be grouped together to create sub-systems. The computational intelligence toolbox, the mechanism modules, the cognitive level and the agent level of the framework all serve to formulate different sub-system architectures for the meta-architecture of the SoS.

**4.2.2. Environment Models.** All sub-system architectures need a physical interface to function. The environment model should capture the Global Information Grid component of the meta-architecture for SoS. Therefore, the environment model plays an important role in models of SoS. Different qualitative and quantitative models can be used to represent the environment model.

Joslyn and Roca (2000) outline a methodology for modeling environments. At the first step, the dynamics of the environment such as physical laws, rules of engagement of

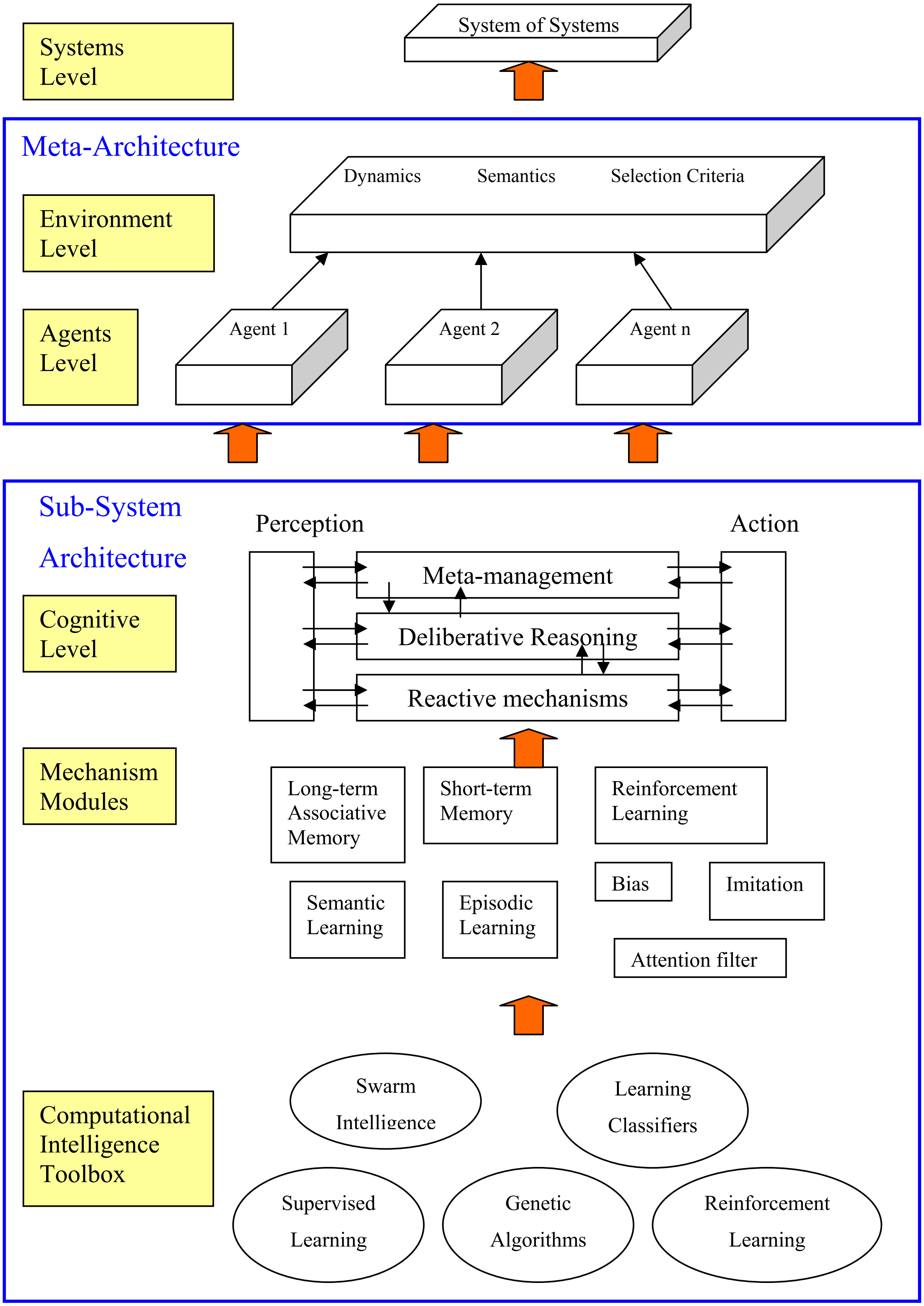


Figure 4.2. The Framework: Cognitive Architecture Embedded in Multi-agent Models

the environment, operational context is specified. These can be natural laws, laws of physics, communication rules, transaction rules, social norms, governmental rules, etc. These rules model the static characteristics of the environment and scope the type of behaviors that are allowed in that environment. Second step is to specify the semantics that agents can utilize while interacting with the environment. These can be artifacts that agents can utilize to communicate the semantics of system laws among themselves. This mainly corresponds to identifying interfaces, which in a way represents the GIG in SoS meta-architecture. The third step is to identify the selection criteria for adaptation. These are the reward system for selecting the successful actions.

The second step of creating artifacts for semantics leads to various models. Agents are modeled within a network of interactions with other agents. Therefore, network theory can be used to model the environment. Network theory utilizes graphs as a representation of either symmetric of asymmetric relations between artifacts. Internet network model, social network models can be utilized to model interactions between agents in an environment.

Parunak (2000) describes environment models as process-interface-topology models. Social norms, governmental rules, weather changes, earthquakes are all processes that can be observed and translated into the environment model. Interfaces can sometimes be only protocol modeling, but for social systems environment interfaces encompass both physical and information influences. Topology model maps processes to other processes through interfaces.

Sometimes a physical environment that imposes constraints on agent’s location can be necessary. This may require spatial models to be incorporated into the environment model. Advanced spatial models can be created by incorporating geographic information systems with agent-based frameworks. Natural system models can also be integrated as artifacts into the environment model. These can be weather models, demographic models, and population dynamics models.

**4.2.3. Meta-architecture Model.** The environment level of the framework and the way the agents are connected to the environment model the meta-architecture of the SoS. The system level of the framework creates an executable model of the meta-architecture, which captures the emergent system level behavior of the meta-architecture.

Modular frameworks that utilize various combinations of architectures and mechanisms promise more flexibility and adaptability for modeling the meta-architecture of SoS.

## 4.3. EVOLUTIONARY ARCHITECTURES

Complex systems architecting of SoS requires the creation of a meta-architecture that consists of core components that remain unchanged for a given period as other components evolved in time. Frameworks for model formulation of SoS should also reflect the evolutionary characteristic of SoS. The framework should provide means for modifying the interfaces between the meta-architecture and other systems that are not in the scope of the original system. The cognitive architecture of the Artificial Life framework plays an important role for evolution of the framework. Sloman et al. (2000) argue that for the H-Cogaff cognitive architecture to be evolvable, a motive generator module is necessary to instantiate a general goal category. This motive generator can be connected to rules of engagement of the environment to update goals of the agent. Once the goals are updated, the meta-management layer can re-arrange lower level mechanisms to reach that goal. Therefore, the meta-management layer of the cognitive architecture plays an important role for the meta-architecture to be evolvable.

More advanced techniques can be used to design evolvable architectures. Independent modules such as large collections of skills, decision-making strategies, short-term memory or attention filter can be reconfigured based on the changing goals. For this, at the meta-management layer of the cognitive architecture fuzzy associative memory can be utilized for assessment of different architectural reconfigurations (Sunghwan et. al., 2001). Another possible evolution technique is to use Genetic Algorithm at the meta-management layer to formulate new processes necessary for the changing environment (Mobley et al., 2006, Hemsathapat et al., 2001). Figure 4.3 and Figure 4.4 illustrate these ideas.

Padberg et al. (2002) propose an object-oriented design methodology for designing evolvable architectures. The initial architecture is composed of loosely coupled components, which contain environment specific rules or general services, such as interact, imitate and learn. All components have the same interfaces: import and export. Import interface contains the services the component uses whereas the export interface contains the services provided by the component. Compositions of components are created to satisfy environmental changes by matching import and export interfaces of components. The meta-management of the cognitive architecture can utilize this type of object-oriented methodology to design new architectures from the mechanism level of the Artificial Life framework.

Another way of designing an evolvable architecture is to use analogies from cognitive science. It is unlikely that babies are born with a fully developed architecture. Therefore, the development of infant cognitive brains can provide insights to designing evolvable systems. For example, it is known that infants initially learn by imitation. Imitation capability can be modeled into the meta-architecture. This capability can allow the meta-architecture to develop new modules by imitating actions of other agents in the new environment.

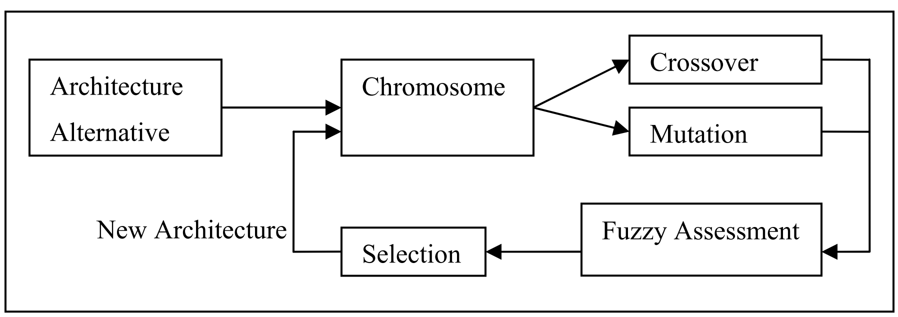


Figure 4.3. Genetic Algorithm for Evolvable Architecture Generation

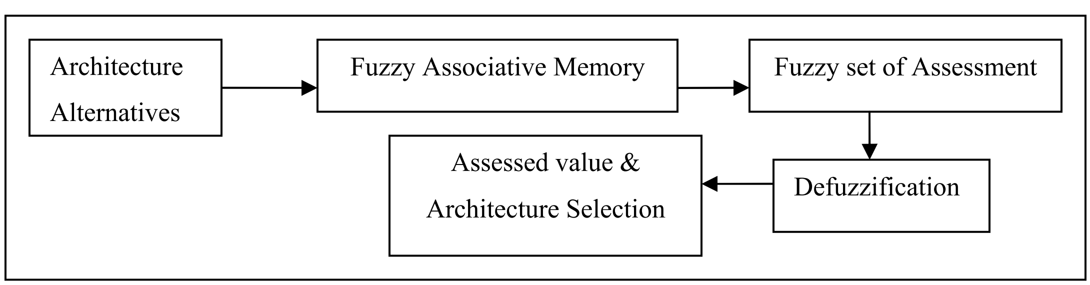


Figure 4.4. Fuzzy Associative Memory for Evolvable Architecture Generation

## 4.4. ADAPTATION OF THE FRAMEWORK TO DIFFERENT SYSTEMS

This sub-section focuses on model formulation for three different systems and illustrates different conceptual architectures derived from the Artificial Life (AL) framework discussed in Section 4.2. The framework is flexible for generating a variety of architectures for modeling these systems. The modular characteristic of the computational intelligence tools and mechanisms allows generation of different sub-system architectures.

**4.4.1. Future Combat Systems.** Future Combat Systems is a SoS composed of eighteen individual systems connected via advanced communications. A soldier linked to these networks and sensors has access to data to gain more accurate picture of what is going on around him. Therefore, Future Combat Systems consist of eighteen individual combat systems, soldier and the network (Johnson, 2003). Figure 4.5 illustrates the Future Combat Systems general framework.

The AL framework can be adapted to create a Future Combat System model composed of three major systems: soldier, manned-systems and unmanned systems embedded in a physical environment and networked to each other via the GIG architecture, which includes networked communications. Three different agent types can be designed for this system. For example, unmanned systems can form flocking behavior by executing simple rules such as don’t move too close to others, match the average velocity of the flock, move towards the center of the flock. This will require a cognitive architecture that will use sensor information and reactive mechanism to select a behavior based on the sensor information. The unmanned system will receive sensor information about others in the flock as well as information about the physical environment they are operating. Swarm intelligence tool can be utilized to design the reactive layer of the cognitive architecture of the unmanned agents. Manned systems are operated by soldiers. These systems can be modeled as reactive systems that operate based on the commands from the human soldiers. In that case, the manned system should be connected to soldier system to receive commands. Soldier system is the most complex architecture that requires a cognitive architecture that has meta-management layer to control reasoning and reactive mechanism, as well as other mechanisms at the perception and action modules of the architecture.

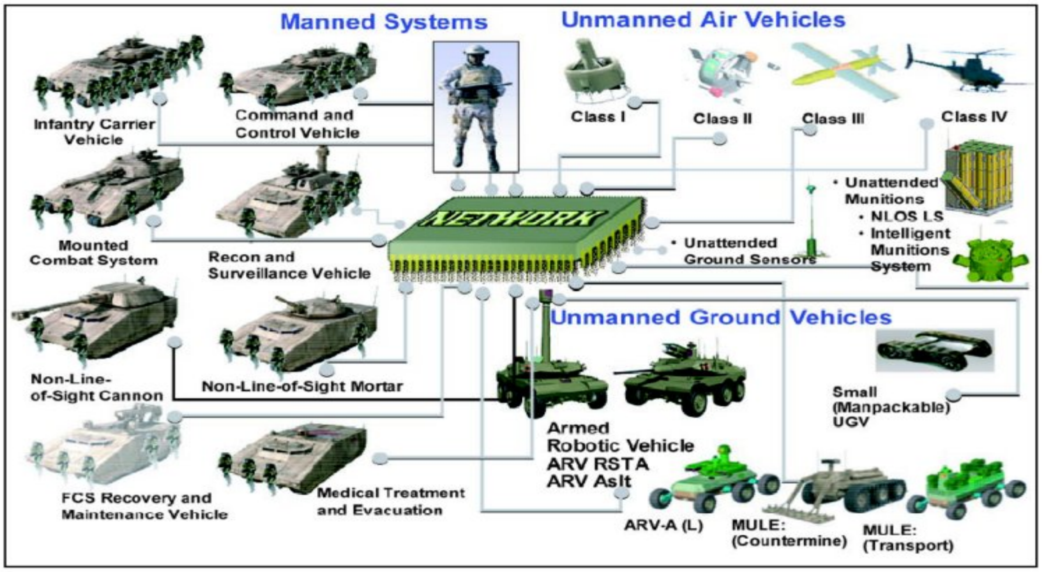


Figure 4.5. Future Combat Systems

Other additional mechanism can be long tem associative memory, attention filter, skills module, interaction module, motive generator module, etc. These additional mechanisms can be incorporated into the meta-architecture based on the abstraction level of the agent-based analysis model. The soldier-decision making process under different operating conditions can be analyzed by utilizing this framework. Scenario generation can be conducted by focusing only on either the unmanned systems or manned systems embedded into the environment models. Figure 4.6 illustrates the main sub-systems of the Future Combat Systems and the sub-architectures derived from the AL framework. All three components are connected to the environment model to form the meta-architecture for the SoS. The environment model consists of a model that represents the GIG, a model that represents the physical environment of the soldier. The environment also comprises the rules of engagement for the agents in the environment.

**4.4.2. Emergency Management.** Emergency management includes emergency operations planning, reporting, resource management and training. The effectiveness of the solutions for emergency evacuation depends on understanding the crowd behavior. AL framework can be used to analyze the crowd behavior during emergency evacuations.

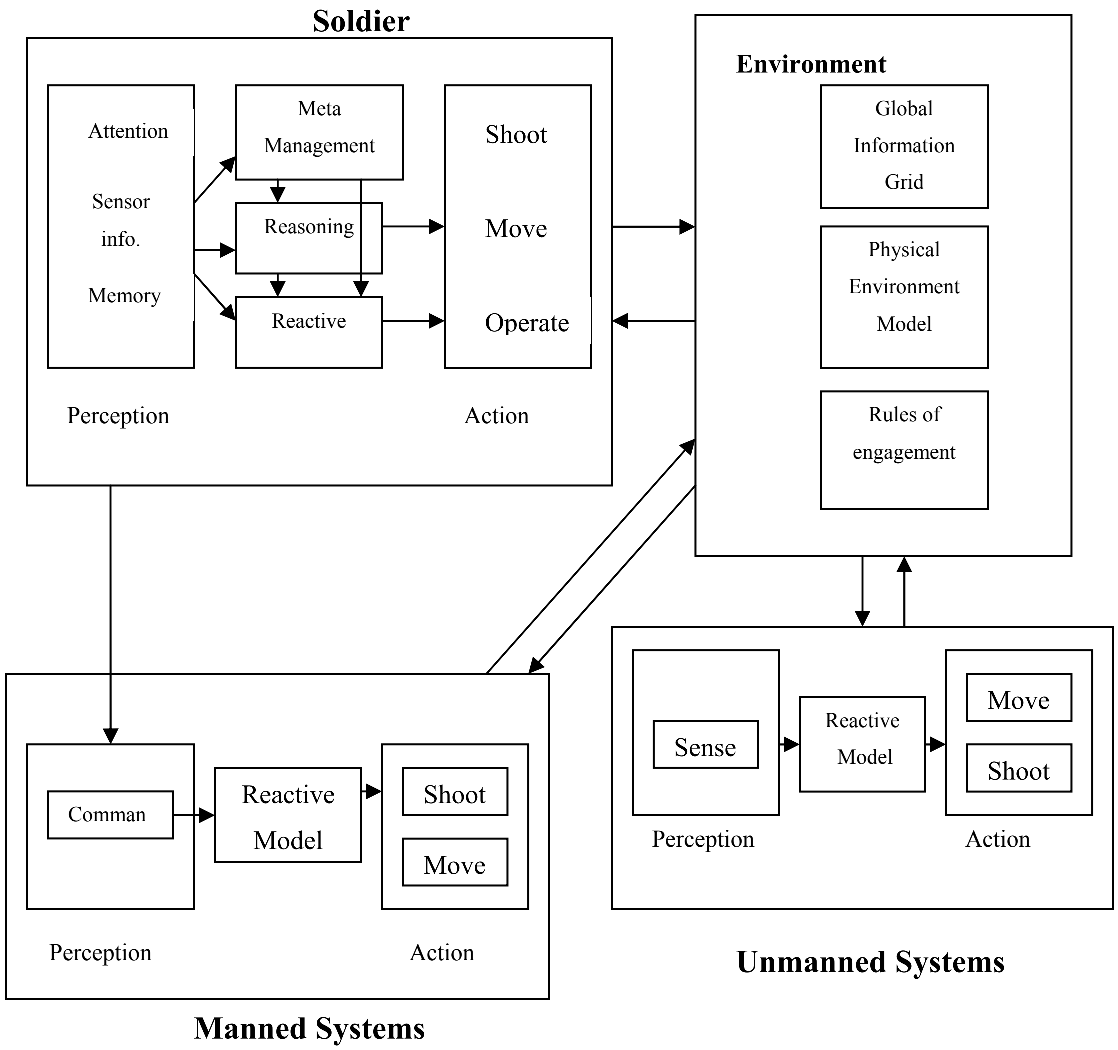


Figure 4.6. Future Combat Systems Analysis Architecture

For example, passengers escaping from a fire in a subway station can be modeled by modifying the AL framework to design a modeling architecture. Summarizing the system, passengers try to escape from fire in a subway station. Subway is a closed tunnel and fire propagates in the tunnel. This forms the environment, which the passengers are embedded. The cognitive architecture for the passenger is composed of a perception module that acquires sensory information as well as crowd influence from its nearby environment. The emotional factors such as panic, fear and impatience cause the passenger agent to lack meta-management and deliberative reasoning mechanisms. Therefore, passenger behavior is generated from simple rules of reactive mechanism. The environment model consists of a spatial model of the tunnel, fire propagation model and smoke diffusion model. Figure 4.7 illustrates the modification of the AL framework for this problem domain.

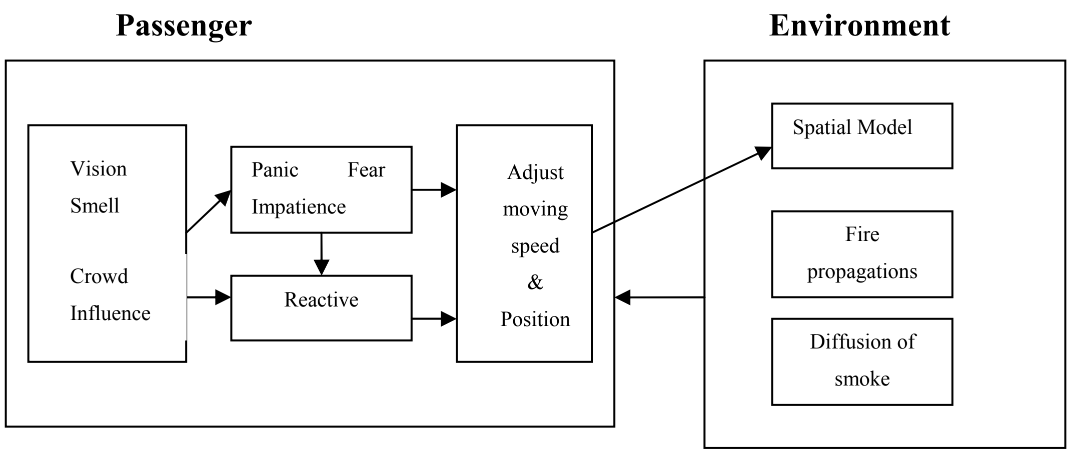


Figure 4.7. Emergency Management System Analysis Architecture

**4.4.3. Global Net-Centric Service Production.** Service production systems are diverse in terms of services they provide. Distribution services can be in wholesale, retail, transport or storage. Knowledge-based services can be in communications, finance or insurance. In-person services can be restaurants, education, health, recreational or government services.  The AL based framework can be used to analyze these systems. The service production selected to modify the framework is a net-centric retail service provider, such as Amazon.com.

Net-centric service producers operate under a network enabled infrastructure environment. Service producer agent receives information about customer requirements, demand. It also receives information from manufacturers about supply of materials. Competitor related information is also used to make deliberative decisions about strategic moves. The deliberative reasoning mechanism of service producers can be modeled using game theory. This can model coordination and competition mechanisms of the service producer. For the customers, the framework can be used to model customer behavior under different social networks. Since the business environment is a strategic environment, deliberative reasoning mechanism consisting of a learning module is enough for modeling both service producer agents and customer agents. Figure 4.8 illustrates this model.

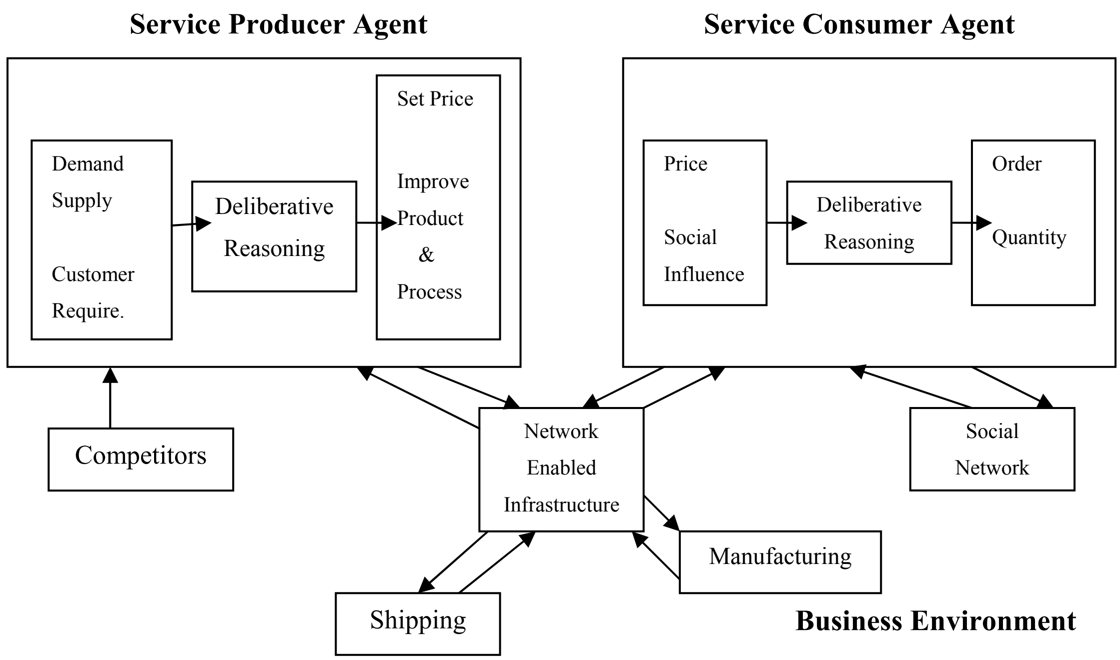


Figure 4.8. Net-centric Service Production System Analysis Architecture

## 4.5. MULTI-METHODOLOGY APPROACH FOR SOS ANALYSIS

The extensive complexity of the SoS architecting requires a multi-methodology approach for analysis of SoS architectures. For modeling SoS, the AL framework described in this chapter can be combined with structural and object-oriented frameworks. A three-step methodology can be used to capture the emergent behavior of

SoS architectures (Kilicay et al., 2007). The SoS architecture is first defined using a structural approach, such as a DoD Architectural Framework (DoDAF). DoDAF defines three related views of architecture development, namely: Operational View (OV), Systems View (SV) and Technical Standards View (TV) (Umheh et al., 2007). These views are used to create a common language for stakeholders to understand the SoS. However, this framework is not sufficient to capture different state models of the SoS. Therefore, at the second step, an object-oriented approach such as UML is utilized to capture the system behavior by identifying end user's requirements, states and sequence of events that the system can undergo (Stanilka et al., 2005). The first two steps still capture the static view of the SoS. Therefore, the third step is to convert the UML static model into an executable model so that emergent behavior of the SoS architecture can be analyzed. Finally, the architecture is modified based on the emergent behavior from the executable model. Figure 4.9 illustrates the multi-methodology approach for analysis of SoS architectures.

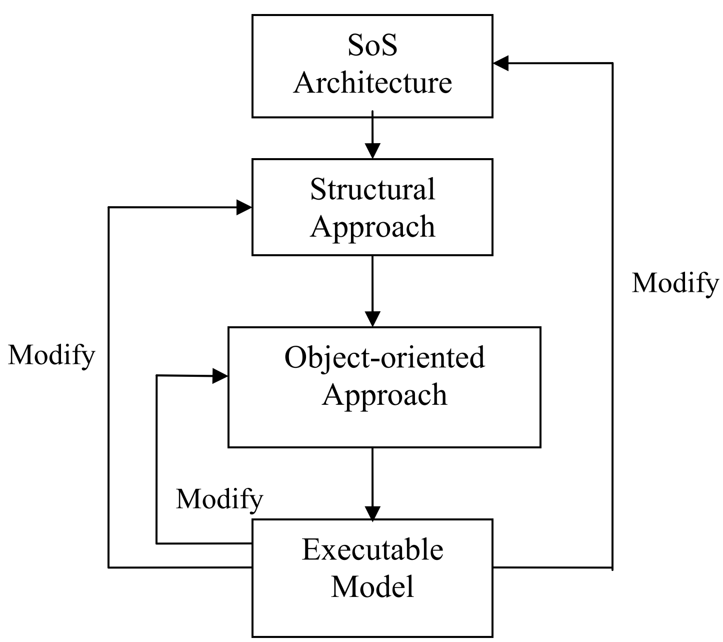


Figure 4.9. Multi-Methodology for Behavior Analysis of Architectures

Petri-nets have been successfully used as an executable model and are easily combined with structural and object-oriented approaches (Madwaraj et al., 2006). The cognitive architecture embedded in multi-agent models can also be used as an executable model for emergent behavior analysis of architectures. These models provide the flexibility to incorporate evolutionary human behavior into system models. This can provide more insights during system architecting. Agent-based simulation packages such as AnyLogic™ have capabilities to convert UML constructs into executable models.

**4.6. SUMMARY**

This section focused on a modeling framework that combines cognitive architectures with multi-agent models. The modularity and the variety of the underlying modules provide flexibility in modeling different SoS at different abstractions. Modular architectures that utilize one or a combination of computational intelligence tools promise more adaptability and robustness for SoS design and analysis. The framework also incorporates human behavioral models through cognitive architectures, which allow analysis of SoS architecture design alternatives under social processes.

Different SAS models can be designed utilizing this framework. This is illustrated through conceptual architectures for three different systems, namely: Future Combat Systems, Emergence Management Systems and Net-centric Service Production Systems.

Seamless integrations and adaptive systems that can respond to changing requirements by reorganizing independent systems are the solution to today’s competitive environment. This characteristic is necessary for both defense and commercial systems and can only be created with evolvable architectures. The framework should also reflect the evolvable characteristic of SoS. Therefore, several methods for creating evolvable architectures are discussed. For this, the meta-management layer of the cognitive architecture plays an important role to create new system architectures that can meet changing goals or environmental conditions.

Finally, one framework is not enough to capture the complexities of the SoS architecting. Therefore, a three step approach that combines structural, object-oriented and executable modeling methodologies are discussed.

Section 5 and Section 6 demonstrate how the AL framework can be used as an executable model to capture emergent behavior of architectures. For this, financial markets are selected as the application system because financial markets show SoS properties. Each trader is an independent system and the market is a collection of these independent systems collaborating implicitly to make profit. Traders play a key role in price formation, but they are affected from the aggregate price changes. They form their future price expectations based on the current price dynamics. Therefore, there is a strong feedback mechanism between the market and the traders. Furthermore, the price clearing mechanism and complex trader behavior create complex relations across different levels of the market. As a result, financial markets exhibit various emergent behaviors such as volatility, persistence of volume, bubbles and crashes. The following sections will focus on capturing the emergent behavior of financial market architectures derived from the AL framework outlined in this section.

# 5. MODELS OF TRADER BEHAVIOR AND ANALYSIS OF FINANCIAL MARKET DYNAMICS

The focus of this section is to capture an understanding of the dynamics of the financial market while demonstrating the Artificial Life framework as an executable model. A combination of tools from the framework is selected to serve this purpose. Learning is a key mechanism in financial markets. Learning classifier systems are selected from the computational intelligence box for modeling the learning mechanism. The reinforcement mechanism exploits successful strategies, whereas the genetic algorithm explores new strategies. This type of learning mechanism is preferred in SoS over supervised learning mechanisms. The cognitive architecture consists of deliberative reasoning layer, which comprises the learning mechanism. The environment model captures the market organization where agents interact only through the price variable. The market organization and the rules for trading at the market create a meta-architecture similar to the SoS meta-architecture described in Section 4.1. Traders are connected to the market trading grid and different system dynamics are observed based on trader behaviors. Figure 5.1 illustrates the financial market meta-architecture derived from the SoS meta-architecture. The following sections provide details of the application study.

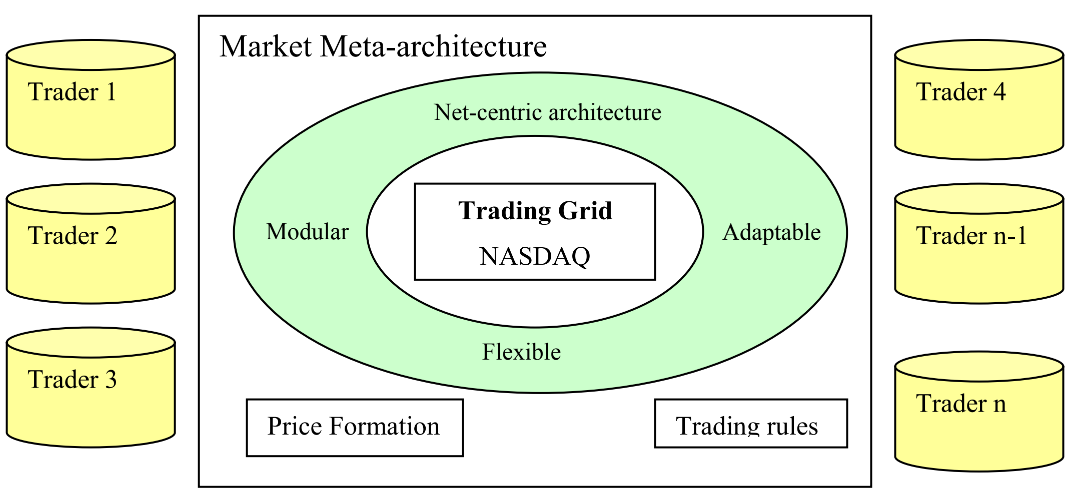


Figure 5.1. Meta-architecture Generation for Financial Markets

## 5.1. INTRODUCTION

The complexity of modeling socio-dynamical environments is a challenge that has drawn attention from varied disciplines. The stock market is also a socio-dynamical system in which governments, corporations and individuals are interested in understanding the main factors that determine market behavior. Therefore, for researchers the focus is on designing models of financial markets to better comprehend underlying market mechanisms. Major paradigms that dominate the study of financial markets include fully rational representative analytical frameworks, behavioral representative frameworks and rationally bounded heterogeneous agent-based evolutionary frameworks (Hommes 2002). Section 5.2 provides a review of the traditional modeling paradigms for analysis of financial markets.

The more recent paradigm is to study financial markets by designing rationally bounded heterogeneous agent frameworks, rather than analytical frameworks. The agent-based approach is an effective tool for modeling and designing heterogeneous agent frameworks. The approach intersects with social sciences and computer simulation fields. Tesfatsion (2001b) provides a survey of agent-based computational economics, including financial markets. Agent-based simulations are naturally suited for modeling market environments because they can define levels of agent autonomy and are able to simulate interactions between investors and the environment (Schoreels et al., 2004). These studies, mainly known as artificial stock markets, apply agent architectures coupled with other artificial life methods in modeling financial markets. In these models, dynamic heterogeneity is critical and this is created by a distribution of agents with a fixed or evolving set of strategies (LeBaron, 2006). The argument behind the shift to agent-based models is that traditional models obtain analytic solutions explicitly and there is no possibility for system behavior to emerge from micro-macro loops. Furthermore, the ideal assumptions of complete information, perfect rationality and common agent expectations lead to predictions that sometimes deviate from observed outcomes in real markets. Financial markets are especially appealing for the agent based approach since investor objectives are clear, financial data are available for benchmarking, and developments in the area of experimental financial market studies provides controlled environments that can be compared with agent-based simulation studies (LeBaron, 2000).

Simon (1969) describes humans as a simple behavioral system where the complexity of human behavior evolves over time as a result of the complexity of the environment in which humans find themselves. Financial markets basically are behavioral systems and it is important to understand the investor behavior evolution under an evolving environment. The study focuses on developing artificial financial market with the followed objectives:

- Incorporate long/short position cover mechanisms into an artificial financial market: The effect of this mechanism on market behavior has not been studied through artificial financial markets. By using discrete state transitions for each trader, the effects of these mechanisms on market dynamics are analyzed.
- Incorporate biased investor behavior: The behavioral models leading to bias in trader decisions and the aggregate effect of various biases to market price formation has not been studied rigorously enough through artificial financial markets. By incorporating biased trading strategies into trader classifiers, the aim is to analyze whether biased strategies survive in an evolving market.
- Design intelligent trading agents: The interest is in studying how traders endowed with learning abilities might co-evolve in societies of learning traders. By using learning classifier systems (LCS), the effect of adaptive learning on price formation is analyzed.
- Analyze the time series properties of the artificial prices: Comparing the statistical properties of artificial time series and return distributions to empirically known properties of real markets validates how much the simulated market resembles real market characteristics.

In Section 5.2, traditional techniques for analysis of financial markets are reviewed. In Section 5.3, related artificial stock market studies are reviewed. In Section 5.4, the artificial financial market model is described. The experimental design and the analysis of the simulation results are given in Section 5.5.

## 5.2. TRADITIONAL MODELING TECHNIQUES FOR THE ANALYSIS OF FINANCIAL MARKETS

One traditional technique for analysis of financial market behavior is formal models (analytical equation-based models) that focus on the relationship with rational traders and asset prices. Another technique which is more recent is behavioral equation-based models that focus on the relationship between irrational biased traders and asset prices. Experimental study with real human traders is also another traditional technique for market behavior analysis. The following sub-sections provide information about these techniques along with their weaknesses.

**5.2.1. Formal Models.** Formal models express relationships among observables using a set of equations. Observables are characteristics or behavior that can be measured (Parunak et al., 1998). The observables are the results of individual behaviors but those individual behaviors are not explicitly represented in equation-based models. Instead, equation-based models use system level observables because it is easier to formulate closed-form equations (Parunak et al., 1998). Equation-based models evaluate the equations over time to produce the evolution of the observables, so model dynamics depend on explicit representation of system level observables.

Formal modeling techniques utilize mathematical description of financial markets to arrive at a formal description of the Efficient Market Hypothesis (EMH). The Efficient Market Hypothesis makes the assumption that securities are always fairly priced and there is no arbitrage opportunity because of competitive pressures among fully informed rational traders. The traditional representative analytical frameworks derive prices from fundamental asset value and models of asset pricing use the Rational Expectations Equilibrium Framework (REEF). REEF assumes individual rationality as well as consistent beliefs. Individual rationality means that traders correctly utilize Bayes' law (which measures uncertainty and degrees of belief as probabilities based on previous experiences and updates beliefs in light of new evidence). Rationality also means that traders' decisions are consistent with Subjective Expected Utility (SEU) which combines trader utility function and trader probability analysis (Barberis and Thaler, 2003). The second assumption that trader beliefs are consistent implies that traders' beliefs are correct because they can process new information correctly and that they have enough information to use the correct distribution for the variables to form their future expectations. The REEF leads to development of rational representative agent models. In these models, a single agent represents the aggregate traders in the market and the model analytically connects the beliefs of this representative agent to asset prices and other variables of the macro-economy. Detailed information for this type of framework can be found in (Fama 1970, 1991).

Therefore, formal models focus on describing financial markets as stochastic processes that describe the evolution of dividends and security prices. The set of mathematical representations of financial markets satisfy the no arbitrage condition of the EMH. Discrete models and continuous-time models are two major formal modeling techniques used to translate the EMH structure into models.

Discrete models assume that there are finite number of states, trading periods and securities. Discrete probability is assigned to these finite numbers of states. Each state is a representation of the evolution of the economy over a period of time. The evolution of the processes of a discrete model can be represented as a path that passes through only one point at a finite number of instants. Therefore, the evolution of financial quantities can be represented as a tree structure. These models assume that phenomena can be described by mathematical models as a function of information available at a given time and allow one to predict the future with some accuracy (Focardi and Jonas, 1997). The challenge associated with these models is the assumption that economy follows a certain determined path through out the entire period which is idealized and not realistic of real markets.

Another version of discrete models is to consider a finite set of instantaneous states for each moment. In these models, the assumption is that there are several well defined paths the economy can follow, but there is uncertainty about the path the economy is actually following. In this case, transition probabilities are assigned at each step as a function of previous step. Instantaneous-state models are known as Markov processes models. Most discrete pricing models are built using Markov processes. The main advantage of hidden Markov models and Markov processes is that they can reproduce empirical phenomena such as volatility clustering. Therefore, these models are combined with other modeling techniques to capture these characteristics of markets. In financial applications, Markov models are used often in volatility estimation and volatility clustering analysis. One challenge in Markov models is that the number of parameters becomes unmanageable when the number of states in the Markov chain increases. For these reasons, small numbers of states are considered.

Markov models provide flexibility in modeling dynamical systems. These models generate a variety of different scenarios. Behavioral finance is focused on the behavioral processes traders take to make a decision. There are a variety of cognitive states traders can take, but the cognitive states the traders will take cannot be identified deterministically. Since these processes are implicit, it is very difficult to fit these states into other forms of mathematical representations. Therefore, Markov models are inherently suitable for this type of behavioral finance problems. These models can link the cognitive states' of traders to asset prices and other market properties. From agent-based modeling perspective, Markov models generate a variety of states and do not limit the system to one type of behavior. Therefore, they can be embedded into agent-based models to capture emergent behaviors.

The other branch of formal models is continuous-time security markets where trading is assumed to be a continuous process and there are an infinite number of states. There are many possible mathematical forms for continuous models. Most popular ones are diffusion processes such as Brownian and Ito processes. Brownian processes are a model for the sources of uncertainty, but not a realistic model for the behavior of securities. Ito processes are stochastic differential equations that capture random fluctuations and deterministic trends of securities (Focardi and Jonas, 1997). Therefore, to model financial markets in continuous-time models, Ito processes are used to represent security price processes and dividend processes, and Brownian processes are used to represent uncertainty. Jump processes and combination of jump and diffusion processes are other continuous- time models used to model financial markets. Continuous-time models are challenging because they require extensive mathematical abstractions and they are not compatible with all possible empirical observations. In fact, all formal models face the challenge of empirical validation.

Another formal model of financial markets is to translate trader behavior into some form of mathematical representation. These models describe traders as investors who maximize a utility function. A utility function is defined over a trader's consumption process. These models are also based on the assumption of market equilibrium. These models represent traders as either decision makers with objective probability distributions or decision makers with subjective probability distributions. The challenge behind these models is that mathematical forms of decision making cannot handle the learning processes of traders. They are also not capable of handling the feedback effects of trades on the market.

The homogeneity in both decision making approach (ideal assumptions of complete information, perfect rationality) and traders having the same common expectations lead to predictions that sometimes deviate from observed outcomes in real markets. Therefore, formal models are sometimes not sufficient enough to represent the full range of possible outcomes from the system when the dynamics are dominated by information processing rather than physical laws.

All formal models are dynamic equilibrium theories which mean that demand and supply are satisfied and there is not any excess demand or supply. Formal models can not handle the price formation mechanism based on actual trading. Therefore, hybrid models that combine the strengths of both modeling approaches should be utilized for better models of financial markets.

**5.2.2. Behavioral Finance Models.** These frameworks depart from rational analytical frameworks by either relaxing the assumptions of individual rationality, or by relaxing the consistent beliefs assumption, referred to as bounded rationality. These models are also generally analytical equation based and one trader represents all traders in the market. The behavioral representative agent drives prices from the biased behavior of irrational investors. Behavioral finance lies on two fundamental facts: the limits to arbitrage and investor psychology. These frameworks explain the behavioral processes the agent takes to reach an outcome, but does not explain the effect of bias under learning processes. Detailed review of this type of framework can be found in (Barberis and Thaler 2003; Shiller, 2003).

One drawback of these models is that these frameworks explain the behavioral processes the agent takes to reach an outcome, but does not explain the effect of bias under learning processes. Also, another criticism of behavioral finance is that individual biases will eventually be priced out of the market (Fama, 1998). However, aggregate bias (social bias) is different from individual bias. Social bias can drive the market out of equilibrium because it creates feedback loops. Agent-based modeling comes into use at this point since it can analyze the aggregate affect of individual biases by incorporating many traders in the model, thereby overcoming this drawback of behavioral finance.

**5.2.3. Experimental and Survey-based Techniques.** These studies conduct experiments with real humans under controlled market settings to analyze trader behavior, information diffusion, and price setting mechanisms. Even though experimental studies have more control on the variables under study, behavior in the laboratory is very narrow in its range (Friedman and Sunder, 1994). Agent-based approach provides an alternative way of expanding experimental studies by incorporating diverse behavioral models. Experimental studies are also criticized by use of biased or unrepresentative sampling (Friedman and Sunder, 1994). Most often college students are used for experiments and the results are generalized to the whole population. Agent-based models can overcome this drawback by providing different population distributions and trader characteristics.

## 5.3. RELATED ARTIFICIAL STOCK MARKET STUDIES

Arguments about market efficiency still continue. Rational frameworks argue that rational agents prevent irrational traders to influence stock prices for long periods through arbitrage, which is an investment strategy that offers risk-free profits at no cost. However, other frameworks argue that strategies designed to correct the deviation from fundamental value can be both risky and costly. An undervalued stock can lose its value more due to bad news about its fundamental value, causing the stock to be undervalued, resulting in arbitrageurs to liquidate their positions early. Besides, transaction costs lower the arbitrage profit, making arbitrage less attractive. Also, arbitrageurs can trade in the same way as the irrational traders and the deviation from fundamental value can survive. Arbitrage is limited if arbitrageurs are risk averse and have short horizons (Barberis and Thaler, 2003). Evidence of persistent deviation from fundamental value can be observed when a stock is added to the index such as the Nasdaq Composite. Its price jumps and much of the jump is permanent. When companies merge or a company sells its subsidiary, persistent under/over valuation can be observed. Barberis and Thaler (2003) provide several historical market examples that show persistent deviation from fundamental values.

Besides these arguments about market efficiency, the aggregate stock market shows several puzzles that traditional models fail to explain. The difference between the real return on risky and risk-free assets, i.e. the equity premium, is difficult to explain. The puzzle is that even though stocks appear to be attractive assets, investors appear very unwilling to hold them and demand a substantial risk premium. Volatility of stock returns and price/dividend ratios is another puzzle that is not well explained by traditional models. Also, the causes of volatility persistence and why markets exhibiting large amounts of trading volume are still not clear. Why stock returns are not normally distributed (fat tails or excess kurtosis) is another issue that lacks further explanation.

Artificial stock market studies can be reviewed by their contribution to understand these puzzles. For example, Pfister (2003) show volatility clustering, fat tails and auto-correlated trading volume by introducing different trading intervals: intraday traders and end-of-day traders. LeBaron (2001, 2002a) approaches volatility puzzle by modeling short-horizon traders and shows that this type of trader increases volatility of the stock significantly.  Chan et al. (1999) develop an agent-based double-auction market which matches the settings of an experimental market with human traders. Their model has investors who forecast price in three different ways: traders who use market information, such as moving averages, to update their beliefs, traders who have fixed strategies and reinforce the ups and downs of price movements, and adaptive investors. In their experiments they focus on information efficiency and deviation from rational expectation price, as well as other issues such as bid-ask spread, trading volume, and wealth distribution among different types of investor. They show that prices converge to equilibrium when the bid-ask spread is narrowing and the volume diminishes. Lux and Marchesi (1999) explain volatility clustering as a result of the changes between number of fundamental and trend traders. The fraction of trend traders is high in periods of high volatility and their analyses show that there is a critical value for the number of trend traders and above this point market destabilizes. They also show that this destabilization is temporary and fundamental traders stabilize the market through arbitrage.

One interesting study that focuses on more than one asset is by Youssefmir and Huberman (1997). They develop a resource allocation environment where agents choose between two resources based on the congestion level of each resource. Their model provided insights to clustered volatility puzzle in equilibrium systems. Their explanation for this behavior is based on the analysis that there may be many forecasting rules that performed well, and that when the system reaches an equilibrium state, agents move randomly and choose from this set of successful rules. This random behavior around compatible forecasting rules can shift a system out of equilibrium, resulting in clustered volatility.

There is no agent-based study that focuses on explaining the equity premium puzzle, but behavioral finance attempts to understand this puzzle by using Prospect Theory, which mainly focuses on systematic violations of Expected Utility (Kahneman and Tversky, 1979) and by using Ambiguity Aversion where people do not like situations where they are uncertain about the probability distribution of a gamble (Heath and Tversky, 1991). LeBaron (2006) points to the fact that agent-based models are also behavioral models because the agents are rationally bounded, but diverges from behavioral finance because of the relatively standard trader preference models. Agent-based market studies, such as Arthur et al. (1996) and Yeh and Chen (2003), model investor preferences as Constant Absolute Risk Aversion (CARA), while studies like LeBaron (2001) model preferences as Constant Relative Risk Aversion (CRRA). One of the few agent-based studies that incorporate behavioral features is by Takahashi and Terano (2002), where they create a financial market to analyze the effect of different investor compositions on the overall asset price fluctuations from the fundamental value. Their market model encompasses fundamental and trend predictors, as well as investors based on Prospect Theory and overconfidence. Their results show that when overconfident investors based on Prospect Theory exist, traded price deviates from the fundamental value. Semet et al. (2004), models traders whose strategies are governed by the estimation of risk currently held by the market. The agent model depends on a risk estimation function, a strategy that maps risk into decision and a price offer for the double auction mechanism developed for the market. Their results converge to efficient market behavior when traders have homogeneous risk preferences and risk threshold. The market deviates from equilibrium when risk thresholds are heterogeneous. They also observe speculative bubbles when risk preferences are heterogeneous and traders are in a panic mode to sell their assets.

LeBaron (2006) categorizes the artificial stock market studies based on their model design as few-type models and many-type models under learning. In few-type models, traders are assumed to choose from small fixed sets of trading strategies and no learning or adaptation is incorporated into internal agent architectures. LeBaron (2000) surveys early influential few-type agent-based markets and provides detailed analysis of agent-based market designs (LeBaron, 2001). In many-type models under learning, traders choose from large evolving sets of trading strategies. Two major trading strategies are seen in these studies: fundamental trading strategies and trend following strategies. Fundamental trading strategies predict the price of an asset by economic fundamentals. Trend followers or technical analysts predict asset prices using simple technical trading rules based on patterns in past prices. The main focus of the many-type models is to understand the effect of learning on market dynamics. For example, the well known Santa Fe Artificial Stock Market (Arthur et al. 1997; LeBaron 1999 and 2002b) explains the statistical properties typically seen in real markets as a result of the learning speed. If trading strategies are allowed to evolve slowly, the market showed behavior consistent with the prediction of traditional economic theory. As the strategies evolved more quickly, the market showed behavior similar to real markets. Joshi and Bedau (1998) expand the Santa Fe Artificial Market study and identify four classes of behavior to explain volatility and large amounts of volume seen in real markets. When there is no evolution, volatility is low and levels of fundamental and technical trading are similar. When the evolution is too fast, complexity of rules is low and prices are not volatile. Slow evolution results in moderately volatile markets where technical trading is low. Fast evolution results in prices being volatile and technical trading strategies dominating.

Market microstructure studies, experimental markets literature, and computational intelligence research provide resources for building agent-based financial markets. Market microstructure studies focus on mechanisms, rules, and structures under which trades takes place, and then analyze the impact of these areas on price formation. Madhavan (2000) provides a comprehensive coverage of the recent literature on market microstructure studies. Gode and Sunder (1993) develop a study to understand the effect of a double auction market structure on overall market behavior. They observe the performance of a double auction trade with human traders and random zero-intelligence (ZI) agent traders. Their study shows an important fact: not all emergent behavior of markets is due to learning and adaptation – some behavior is due to the structure of the markets. Marchesi et al. (2000) incorporate a market maker who matches offers and demands and has unlimited availability of cash and stocks satisfying all orders. In their model, the price formation is given by the intersection of the supply and demand curve. Their artificial time series show fat-tail properties of real markets. Dermietzel et al. (2006) compare different market clearing mechanisms and their influence on prices. They use Walrasian Adaptive Simulation Market (WASIM), which is a generic model build on the Santa Fe artificial stock market, to compare the prices determined by a Walrasian auctioneer to prices determined by market makers. Their results show that automated market maker is able to approximate equilibrium prices with linear price adjustments to excess demand/supply. Human market makers are more appropriate in high volatile markets and a Walrasian auctioneer guarantees a price close to equilibrium but requires more information, including the wealth of agents, trading restrictions, and supply/demand functions of all agents.

Experimental market studies are an alternative approach to the theoretical microstructure approach. This field conducts experiments with real human traders under controlled market settings. Experimental studies are successful in analyzing ultimate investor behavior, but the dynamics of investor behavior, such as learning and heterogeneous preferences, are not modeled explicitly (Poggio et al., 2001). The experimental studies provide important information for validation of agent-based models.

Computational intelligence studies provide various models of learning and internal agent architectures. Besides Genetic Algorithms (GA), the field is abundant with different learning models, such as neural networks, reinforcement learning algorithms, learning classifier systems, fuzzy logic, and evolutionary swarm techniques. Many models, including the Santa Fe Artificial Stock Market studies, incorporate learning using learning classifier systems (LCS). There are two types of LCS depending on where the genetic algorithm acts. In a Pittsburg-style LCS, the genetic algorithm acts on a population of separate rule sets. In a Michigan-style LCS, the genetic algorithm acts on only a single population. Michigan-style LCSs have two types of reinforcement learning; fitness sharing (ZCS) and accuracy-based (XCS). Sculenburg and Ross (2000) use Michigan-style learning classifier systems to evaluate several types of agents receiving different environmental messages. They provide historical real stock prices to the agents with the aim of exploring reliability of artificial traders under real economic environment. They analyze the affect of market on the trader strategies instead of analyzing the effect of traders on price dynamics. Their experiments show that learning classifier systems are able to represent competent traders where agents are successful in finding profitable rules and where technical trading is a valid outcome in markets. LeBaron (2001) utilizes neural networks combined with genetic algorithms and represents trading rule sets as a function, mapping past information into current portfolio weights. Tan and Lin (2001) use fuzzy logic to model expectation formation. Chen (2003) deals with a multi-agent-based architecture for artificial stock markets and incorporates a public place for social learning. Some agents use the public forecast models or trading rules to make decisions, whereas some investors only use their own private forecasting models and trading rules. When investors that use their own individual strategies are unsatisfied with their wealth, they can also access the public base. Different from other learning-based artificial markets, genetic programming is used to model the cognitive behavior of agents. Genetic operations are applied for maintaining a diverse evolving forecast model or trading rule population. Incorporation of a public place for social learning results in a rapidly changing market environment where the value of a successful trading rule or forecast models depreciate at high speed. Ultimately, their analysis shows that there is no significant tendency of trading rules or forecast models to get more complicated.

Agent based models broaden financial market studies by integrating theoretical models (from microstructure) and internal agent models (from computational intelligence) into one analysis system. The study expands experimental studies by adding learning models, asymmetric information structure, different behavioral investor characteristics, and heuristics into trading strategies. Variations in trading mechanisms, determination of market price, types of assets or securities, and investor behaviors result in different market behavior.

## 5.4. ARTIFICIAL STOCK MARKET MODEL

The market structure consists of several major parts: agent types, agent trading rules, securities, price formation, and evolution. In the following sub-sections, some of the assumptions regarding the key market components are discussed.

**5.4.1. Security.** The security structure of the market is relatively simple since the focus is on investor behavior under changing market outcomes. Therefore, there is only one stock for trading and this stock pays a dividend. Since dividends are usually deterministic, uniform distribution is selected to determine the dividend process. A risk-free T-bill rate is revealed to every trader so that they can form their price expectations based on a benchmark.

**5.4.2. Agent Types.** The system is based on the concept of simulating traders in an equity market, with each agent having an initial bias about the trading decision they make. At the initial trading step, a predetermined percentage of the total agent population is set as long traders, short traders, and traders who take no position. As trading progresses, investors shift their bias based on their trading strategy, resulting in price changes. Investors take one of three different decisions:

- <u>Long Position</u>: Buy the stock at the current price and then sell it at a higher price to make profit.
- <u>Short Position</u>: Sell the stock at the current price. When the stock price falls, buy the stock at the low price (cover short position) to make profit.
- <u>No Position</u>: Do not trade

Heterogeneity of agents results from their initial trading bias, as well as the profit margins they use to cover their position. The profit margin of each trader is directly related to the trader's price expectation. For example, if the trader has a long position and expects that the stock price will increase by 5% in the next trading period, then the profit margin to cover the long position is also going to be 5%. This profit margin is a variable in the classifier system of the trader and evolves based on the market conditions. For this study, the internal trader bias mechanism is not modeled explicitly because the focus is on understanding the external dynamics leading to price changes. Figure 5.2 summarizes the trader behavior. Trader behavior is activated by the XCS learning classifier system, which will be explained in the following sub-sections.

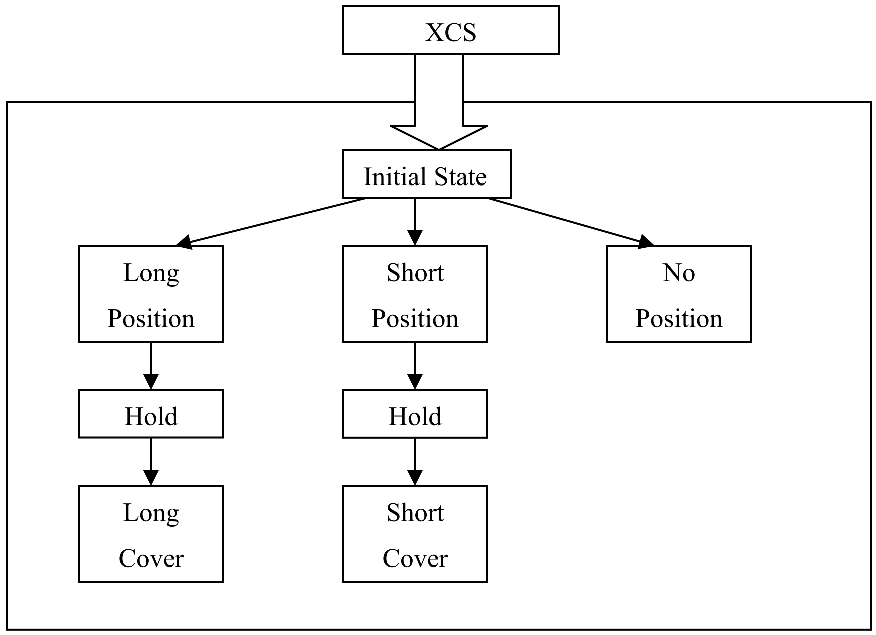


Figure 5.2. Trader Architecture/Behavior

**5.4.3. Agent Trading Rules.** For this simulation, technical trading rules are used as trading strategies for agents (Kilicay et al., 2005, 2006b). There are no traders that calculate the fundamental value of the stock. Traders look at the past trading periods and make their decisions based on past trends. Types of technical rules used in this simulation include:

- Moving Average (MA): The MA shows the average value of a securities price over time (Schoreels, 2004). Here, MA(10) uses 10 periods for the calculation and is used as a short-term moving average, while MA(20) is calculated for 20 periods and is used as a long-term moving average indicator. The moving average is calculated as:

$$M(t) = \sum P(t) / N \quad (1)$$

where for this simulation N=10 for MA(10) and N=20 for MA(20). The decision rules for agents using MA(10) are listed in Table 5.1.

Table 5.1. Decisions Based on the MA Indicators

| **Condition** | **Decision** |
|---|---|
| Price > MA(10) | Long position |
| Price < MA(10) | Short position |
| Price = MA(10) | No position |
| MA(10) > MA(20) | Long position |
| MA(10) < MA(20) | Short position |
| MA(10) = MA(20) | No position |

- Rate of Change (ROC): The ROC indicator is based on the assumption of cyclical price movements and considers the relative change of prices over time to indicate trends (Schoreels, 2004). ROC is calculated as:

$$ROC = P_{(t)} - P_{(t-N)} \qquad (2)$$

where N=5 for short-term analysis and N=10 for long-term analysis. Some of the decision rules for agents using ROC are listed in Table 5.2.

Table 5.2. Decisions Based on the ROC Indicators

| **Condition** | **Decision** |
|---|---|
| ROC(5) < 0 | Long position |
| ROC(5) > 0 | Short position |
| ROC(5) = 0 | No position |

- Volume: Volume (Vol) is the total number of shares traded in each period and often provides useful information in trading decisions by way of validating the strength of a price move. Traders in this study will use volume as another indicator to make their decisions. A volume of 5 periods will be used for short-term and 10 periods for long-term traders when making a decision. Table 5.3 lists the decision rules based on the volume indicator.

Table 5.3. Decisions Based on the Volume Indicator

| **Condition** | **Decision** |
|---|---|
| Vol(5) > Vol(10) | Long position |
| Vol(5) < Vol(10) | Short position |
| Vol(5) = Vol(10) | No position |

A variety of trading rules can be generated by combining various indicators. For example, the volume indicator can also be combined with MA indicators. The Relative Strength Index (RSI), Moving Average Convergence Divergence (MACD), Stochastic Oscillator, and Bollinger Bands are other commonly used technical analysis indicators, but are not used by traders in this simulation.

Besides technical trading rules, traders also have several biased trading strategies. The biased trading strategies are derived from the trader bias model proposed by Barberis et al. (1998). Their model shows two types of bias seen in humans: representativeness and conservatism. Representativeness is the tendency of humans to see events as representative of some class and think they see patterns or trends in random sequences. Conservatism is the tendency of individuals to change their beliefs slowly under new information. Traders form their future expectations based on changes of dividend earnings. In Barberis's model, traders believe that dividends move in two Markov states. Traders believe that a positive dividend earning will be followed by a positive dividend earning in the next period with a certain probability. A conservative trader's probability is between pc = 0-0.5, and they believe that a positive shock is likely to be reversed in the next period. A trend follower trader's probability is between pt = 0.5-1, and they believe a positive shock is likely to be followed in the next period. The probabilities in both states are fixed in the trader's mind. Each period, when the dividend is revealed, the trader uses this information to update beliefs about the current market state. Based on this model, some of the biased strategies are provided in Table 5.4. Δd represents the change in dividend.

Table 5.4. Biased Decision Strategies

| **Condition** | **Decision** |
|---|---|
| If Δd >0 and p <pc (conservative) | Long Position |
| If Δd >0 and p >pc (conservative) | Short Position |
| If Δd >0 and p <pt (trend follower) | Short Position |
| If Δd >0 and p >pt (trend follower) | Long Position |

Traders still have technical trading strategies that evolve based on market dynamics, but they also have biased trading strategies that interrupt their learning mechanism.

**5.4.4. Price Formation.** Response to excess demand determines the price (P) of the stock. First, each trader determines the expected future price based on the signals from their classifier system. Then, based on the expected price, the trader determines the number of shares for trading. All traders share the same constant absolute risk aversion (CARA) utility function (Arthur et al. 1996; Hommes 2002):

$$U(W) = -\exp(-\lambda W) \tag{3}$$

where W is the wealth of trader and $\lambda$ is the risk aversion. Therefore, determination of risky asset amount is independent of investor wealth. In reality, wealthier traders have a greater impact on the prices, and CARA ignores this fact. Investors have two assets, risk-free T-bills and stock shares. The trader's wealth can be expressed as:

$$W = T + (P)x \tag{4}$$

where T is the money from T-bills, P is the stock price, and x is the number of shares. Traders' wealth in the next period is then:

$$W\ (t+1) = (1+r)\ T + (x)(P_{t+1} + d_{t+1}) \tag{5}$$

where r is the risk-free return. Each trader myopically maximizes the one-period expected utility function,

$$E_t\ (U\ (W_{t+1})) = E\ (-\exp\ (-\lambda\ W_{t+1})) \tag{6}$$

Under CARA utility, traders demand for the risky asset is given using equation 7:

$$x_t = \frac{E(P) - (1-r)P_t}{\lambda \sigma^2} \tag{7}$$

where $x_t$ is the demand/supply by the agent at time t, $E(P)$ is the expected price prediction made by agent, $P_t$ is the stock price at time t, $r$ is the risk-free rate of return, $\lambda$ is the degree of risk aversion and $\sigma^2$ is the variance of the expected stock price.

Market demand (D) and supply (S) from each trader are summed, and if there is excess demand, the price of the stock is increased by α amount (Lettau, 1997). If there is excess supply (S), the price is decreased by α amount. Equation 8 summarizes this pricing mechanism:

$$P_{t+1} - P_t = \alpha\ (D_t - S_t) \tag{8}$$

**5.4.5. Evolution.** Holland's classifier system has been used extensively in modeling learning mechanisms in artificial financial markets (Schulenburg et al. 2000; LeBaron, 2002) due to its ability to tackle high dimensional state spaces. Learning classifier systems have three main components: the performance component, the reinforcement component, and the discovery component. Trading rules classify the states of the environment into categories in the form of condition (if) – action (then) form. The conditional part of each rule consists of a string of symbols {0, 1, #} which are matched against the current market state - "1" matches the market condition, "0" does not match, and "#" means do not care.

Signals from the environment, in the form of strings, are received by the classifier system and several rules whose conditions satisfy environment conditions compete for final execution (Lettau, 1997) at the performance component. At the reinforcement component, successful rules are rewarded, while at the discovery component, the genetic algorithm (GA) is applied for discovering new classifier rules and eliminating unsuccessful ones. Genetic Algorithms are powerful tools for simulating evolutionary processes, including economic learning. The GA covers different regions of the search space. Too much exploitation can result in missing the global optimum, but too much exploring hurts system performance. Therefore, balance between exploration and exploitation is an important component of adaptation and learning systems. The GA balances the trade-off between exploration and exploitation. Vriend (2000) analyzes the two major processes related to learning: individual and social learning. A change in the perception of the environment results in individual learning whereas a change in the environment itself is social learning. Therefore, identical learning algorithms lead to different results when applied as a model of individual learning and when applied as a model of social learning. There are two basic ways to implement a GA:

- <u>GA as a model of social learning:</u> Each individual imitates successful individuals. This is represented such that each individual in the population is characterized by an output rule.
- <u>GA as a model of individual learning:</u> Instead of being characterized as a single output rule, each individual has a set of rules where each rule has a fitness measure and at each period one of the rules is used.

This study also utilizes a learning classifier system as an individual learning mechanism, but instead of Holland's classical classifier system (Bull, 2005), Wilson's XCS based classifier system is used (Wilson, 1995). This selection was necessary because classical classifier systems use the strength parameter as a predictor for future payoff, as well as the classifier's fitness for the genetic algorithm. However, a low predicting classifier can be the most suitable choice for its environmental niche. Also, since the fitness is based on predicted payoff, the system can eliminate a useful classifier. Furthermore, in classical classifiers the guesser type classifiers (classifier string that has “#” character in all of its positions) will be encouraged because the GA cannot distinguish an accurate classifier from a general classifier since each have the same payoff (Wilson, 1995). The XCS-based classifier provides solutions to these problems and thus is selected to be a better model of learning for this study.

Figure 5.3 illustrates the XCS Learning Classifier System. Here the GA is applied to the match set instead of applying it to the population set in classical systems. The reward is also given only to the action set instead of giving it to match set in classical classifier systems. Detailed information about XCS classifiers can be found at (Wilson 1995; Butz 2002; Bull 2005). For this study, classifier condition part is represented as strings of length five with each position taking any values of “0”, “1” and “#”. Initially, the rules are generated randomly. They are a mapping from states of the market into actions and forecasting parameters. States of the market environment are represented by the technical indicator values mentioned in the agent trading rules section. The action part of the classifier system is represented as string of length two, where the first position represents a buy (1) or sell (0) signal and the second position represents the expected percentage price change of the trader. For example, a classifier string might look like the following: 1, 0, #, #, 0: 1, 5%. This classifier means that current price >MA(10) is true, current price >MA(20) is false, do not care about ROC(5)<0, do not care about ROC(10)<0, and V(10) < V(20) is true. If these conditions hold, then take a long position with an expected price increase of 5%. All relevant learning parameters for the XCS, as well as the other experimental settings, are chosen similar to (Butz et al. 2002). Table 5.5 provides some of the relevant parameters.

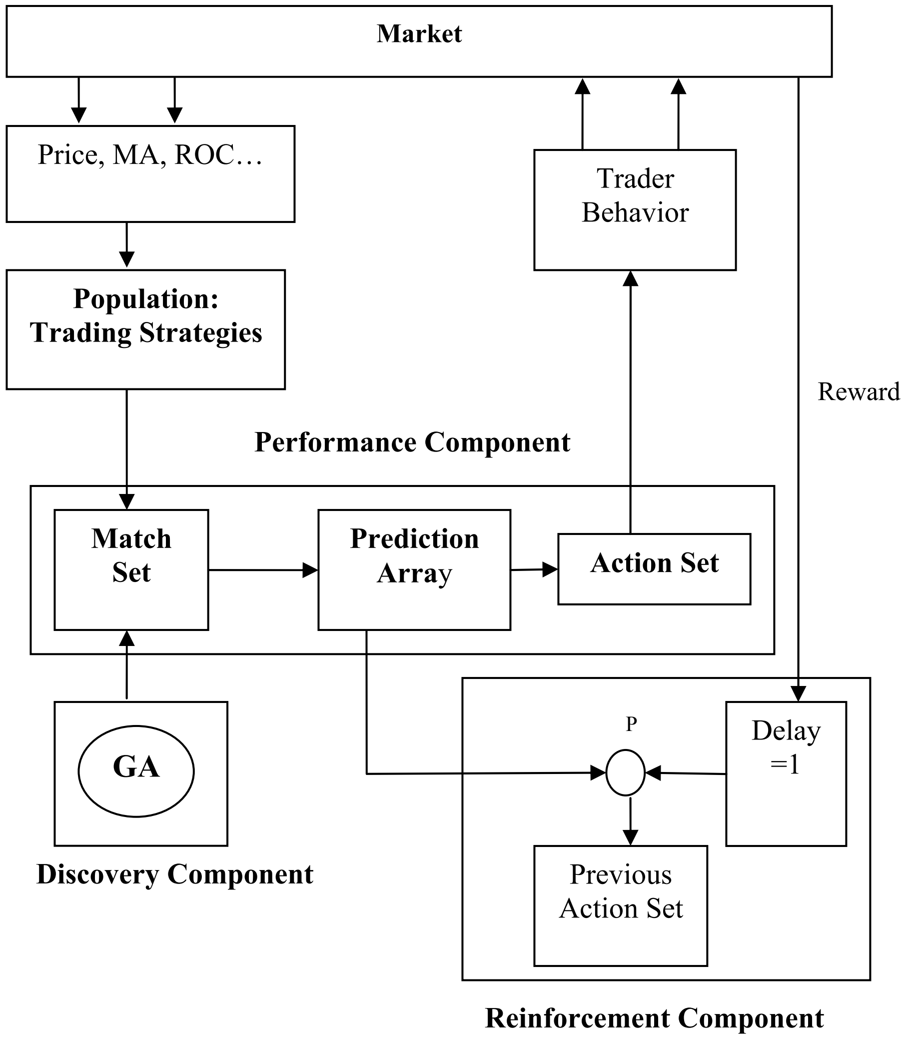


Figure 5.3. Learning and Evolution Architecture: XCS Learning Classifier System

Table 5.5. Selected Parameters for the XCS

| | |
|---|---|
| Learning rate for updating fitness, prediction, prediction error | 0.2 |
| Crossover | Two points |
| Probability of crossover | 0.8 |
| Probability of mutation | 0.04 |

## 5.5. EXPERIMENTS AND RESULTS

**5.5.1. Simulation Architecture.** The simulation starts with an initial population of long traders, short traders and traders that do not take any position. This is necessary to create an initial price series. Traders, based on their behavior type, determine their demand/supply for the shares. Market price is determined from the aggregate demand/supply of traders. Once price is revealed, the market calculates and stores several technical indicators such as moving averages and trading volume. In the next trading period, the XCS mechanism for each trader starts running and gets the current market technical indicators. The system selects an action from the trader's trading rules and this action fires a specific trader behavior. The system rewards trading rules if the trader is successful in covering their position. Figure 5.4 shows the simulation architecture and the relationship between micro and macro loops.

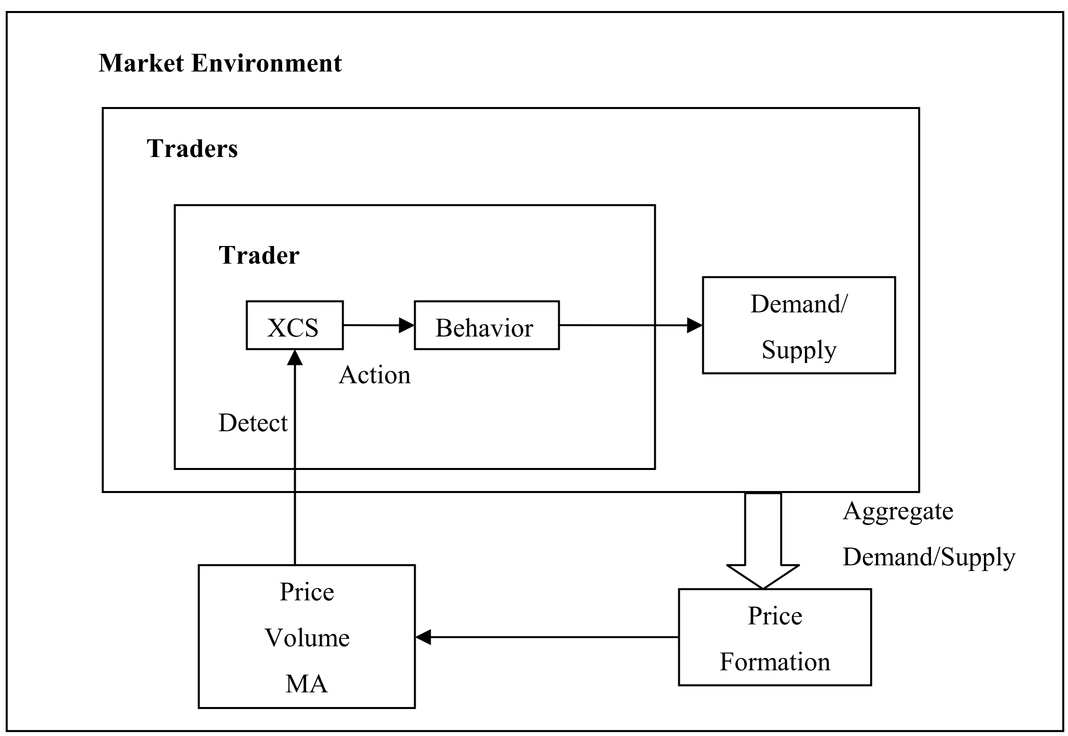


Figure 5.4. Market Architecture

**5.5.2. Model Parameters.** Table 5.6 summarizes the parameters used in the simulation. Since the aim of this study is to understand effect of various mechanisms on price dynamics, the parameters are not adjusted to create time series that reflect real markets. Several parameters use values seen in real markets, but this is only used to assign initial values. For example, for initial price and volatility, historical 2005 values for a multi-national technology firm stock are used. The risk-free rate is selected from 2005 risk-free rate. The total number of stocks traded is an important parameter for maintaining system equilibrium. The risk aversion value is selected from Takahashi and Terano (2002). The total amount of traders is set as 100 agents so that various trader types can be sufficiently represented. The model has the flexibility of increasing the number of traders in the market to see if total trader size has any effect on the market outcome. Time needed to cover a position is selected from a uniform distribution. The range is selected to allow traders enough time to check the market price and take a position to cover. The traders are not allowed to take a different position until they check for covering. If they are unable to cover in that time period, they miss their position. Table 5.6 shows the parameters used in simulation experiments.

Table 5.6. Simulation Parameters

| **Parameter** | **Description** | **Value** |
|---|---|---|
| N | Number of traders | 100 |
| S | Total number of stocks | 1100 |
| $\lambda$ | Risk aversion | 1.25 |
| rf | Risk-free interest rate | 0.04 |
| $\sigma^2$ | volatility | 0.12 |
| T | Time for covering | Uniform(0.2-0.5) |
| Tr1 | Initial number of long traders | 45 |
| Tr2 | Initial number of short traders | 45 |
| Tr3 | Initial number of no position traders | 10 |

**5.5.3. Experimental Designs and Simulation Results.** Simulation experiments are generated using the Anylogic5.1 simulation software™. It is a hybrid multi-paradigm simulator capable of modeling systems as a combination of discrete-event, systems dynamics and agent-based models. Therefore, it is suitable for simulating complex, dynamic heterogeneous systems. Detailed information and demos of the software can be found in http://www.xjtek.com/. XCS implementation in Java is downloaded from Illinois Genetic Algorithms Laboratory (IlliGAL), University of Illinois at Urbana-Champaign, which can be found in http://www-illigal.ge.uiuc.edu/sourcecd.html. This code is modified and combined with the Artificial Stock Market developed in Anylogic5.1™.

Five replications of the same simulation are conducted to observe if the results are consistent from replication to replication. Each simulation takes 1000 trading periods. The replications revealed the same results. In order to understand the effect of various mechanisms, five different scenarios are considered. Table 5.7 shows the differences between these markets. In Market A, no learning or covering mechanism is incorporated. Market B has a learning mechanism but still no covering mechanism. The comparison between Market A and Market B will give insights about the effect of the learning mechanism on the market. Market C and D incorporate covering mechanism: Market D has learning mechanism; where as Market C does not. Comparison between Market C and D will also provide insights about the effect of learning under a different market structure. Comparison between Market B and Market D will provide insights about the effect of covering mechanism on price dynamics. Finally, Market E is designed to understand whether biased trading strategies have any aggregate effect on the market behavior and whether these strategies survive under a learning mechanism.

Based on these different designs, simulations are conducted using the parameters shown in Table 5.6. The time series graphs provide information about several questions of interest.

The effect of covering mechanism on market behavior:

Comparison of Market B and Market D reveals some information about the effect of the covering mechanism. Figure 5.5 shows Market B price formation. Figure 5.6 shows Market D price formation from one run of the five replication runs.

Table 5.7. Scenario Generation

| | **Cover mechanism** | **Learning** | **Biased strategies** |
|---|---|---|---|
| Market A | No | No | No |
| Market B | No | Yes | No |
| Market C | Yes | No | No |
| Market D | Yes | Yes | No |
| Market E | Yes | Yes | Yes |

As seen in Figure 5.5 and Figure 5.6, the price fluctuates between 75-105 dollars in Market B when there is no covering mechanism, whereas the price fluctuates between 70-80 dollars in Market D. As the price increases in Market D, long traders begin to cover their positions and start selling their shares at a higher price, resulting in price to drop, clearly illustrating the covering mechanisms. The covering mechanism creates a cyclic form of time series where the price fluctuates between upward movements followed by downward movements. The relatively calm behavior under covering mechanism can provide insights about why markets fluctuate between turbulent and calm periods. Under normal conditions, where traders have access to similar type of information and decision making methods, the covering mechanism forces the system to stay in calm periods. Other factors, such as trader bias can drive the system into turbulent movements. Market E design provides information about the effect of biased trading strategies on market dynamics and will be discussed in the following paragraphs.

<u>The effect of learning mechanism on trader behavior and market outcomes:</u>

Comparison of Market A and Market B, as well as comparison of Market C and D reveal some information about the effect of learning on market dynamics. There is no learning mechanism in Market A and Market C. Figure 5.7 shows price formation Market A and Figure 5.8 shows price formation in Market in Market C. In Market A, where a covering mechanism does not exist, the price series is in an upward trend. Since trader behavior is random, there is no mechanism to stabilize the system. In Market C, prices are relatively calm due to the covering mechanism.

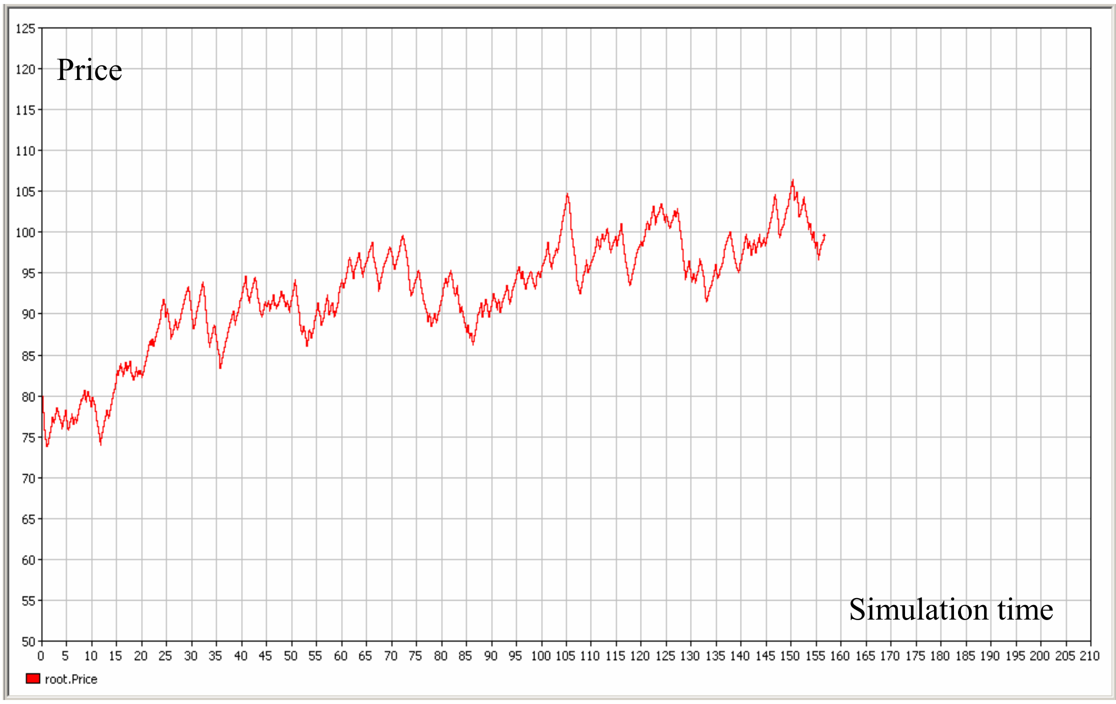


Figure 5.5. Price Formation in Market B

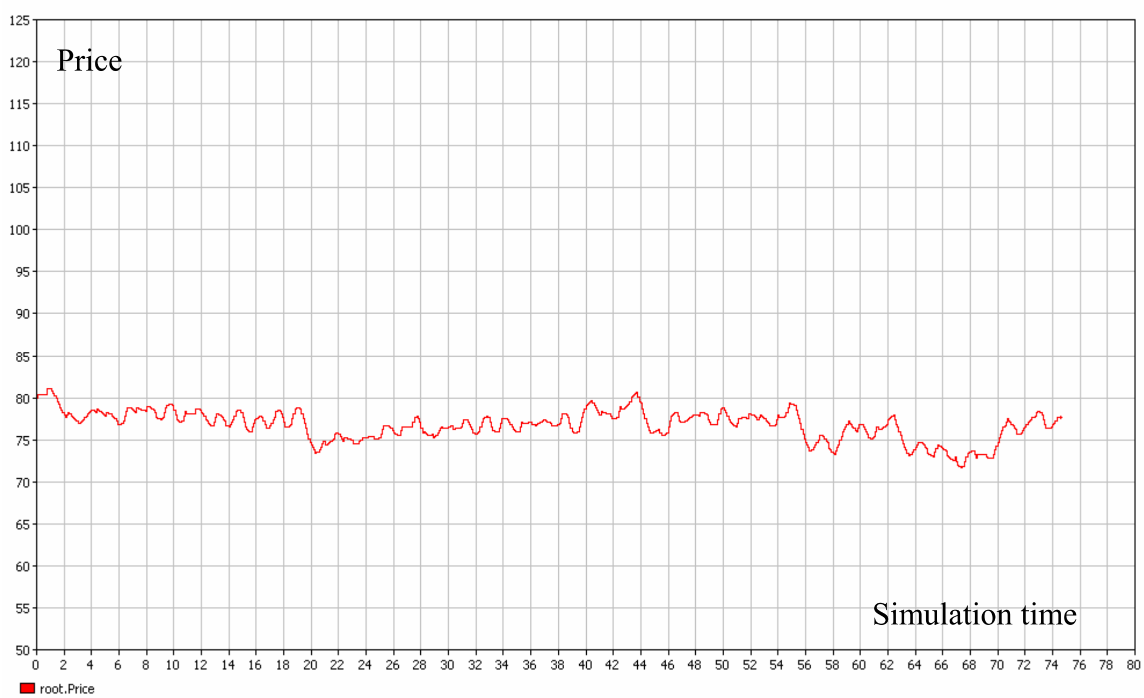


Figure 5.6. Price Formation in Market D

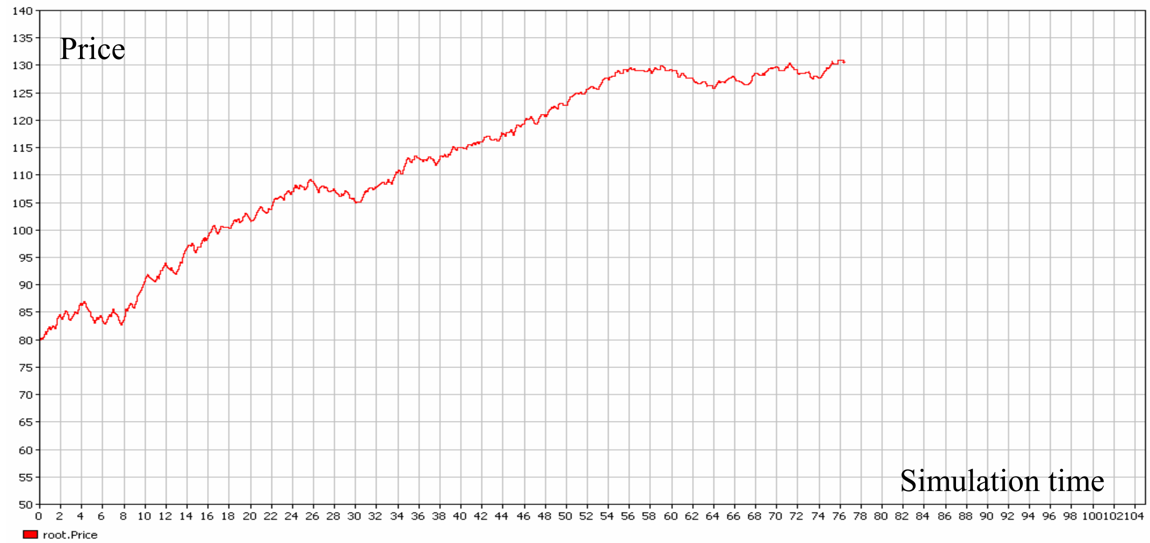


Figure 5.7. Price Formation in Market A

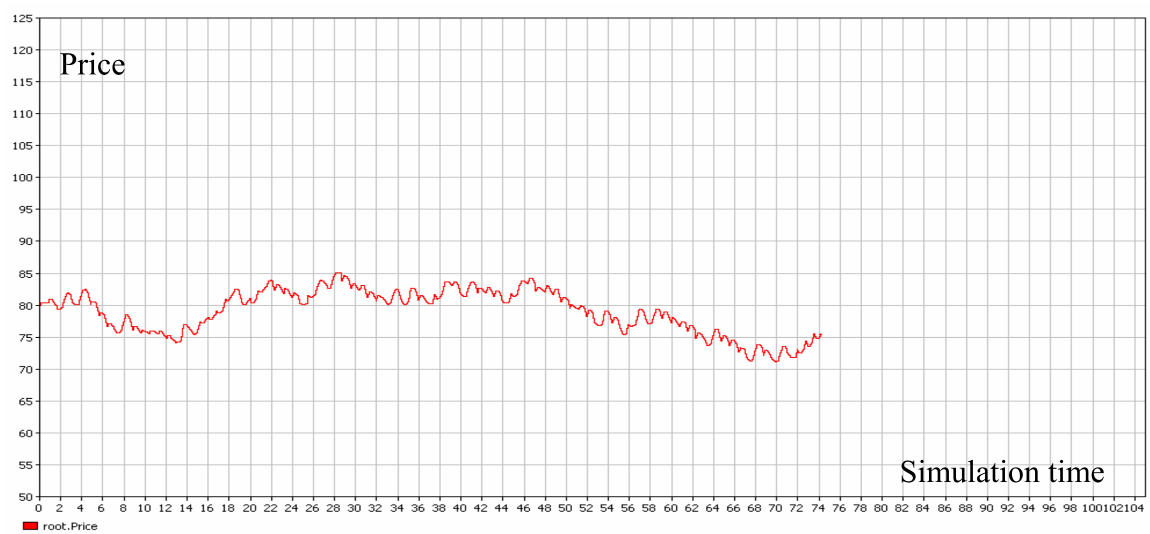


Figure 5.8. Price Formation in Market C

It is also useful to understand which trading strategies dominate as the learning progresses. Table 5.8 provides the representation for each classifier positions. Table 5.9 shows the commonly fired strategies and actions in Market D at the end of 1000 runs.

Table 5.8. Classifier Positions

| Position 1 | Position 2 | Position 3 | Position 4 | Position 5 |
|---|---|---|---|---|
| If P > MA (10): 1 else : 0 | If P > MA (20): 1 else : 0 | If ROC (5) > 0: 1 else: 0 | If ROC (10) > 0: 1 else: 0 | If Vol (5) > Vol (10): 1 else: 0 |

Table 5.9. Evolution of Trading Strategies in Market D

| Classifier | Action |
|---|---|
| ##1#0 | 2 |
| ###0 | 1 |
| #00## | 3 |
| ##0## | 0 |
| #11## | 4 |

In market D, the strategy using Moving Average (10) indicator is eliminated in the sample simulation runs. The strategy using ROC (10) is also eliminated. As the simulation progresses, traders utilize classifiers that have ROC (5), MA (20) indicators or volume indicators. There are five different actions the traders can take. The main difference between the actions is the percentage amount of price increase or decrease traders expect to observe in the next trading period. This value is also the profit margin range traders use to cover their positions. At this point of the simulation, traders utilize all actions.

In Market B, where there is no covering mechanism, the trading strategies evolve towards more generalized strategies. Strategies that utilize ROC (5) and ROC (10) indicators dominate trader decision making. Other strategies, such as MA (10), MA (20) and volume indicators are eliminated from the system. Table 5.10 shows the most commonly fired trading strategies and actions in Market B at the end of 1000 runs.

Comparing the evolution of trading strategies in Market B and D show that technical trading strategies are more successful in markets where covering exists.

The effect of biased trading strategies on market outcomes:

Figure 5.9 shows the price formation in Market E where two of the technical trading strategies are replaced by the two biased trading strategies explained in Section 5.4.3. Even though covering mechanism exists in Market E, the price fluctuations are higher than Market D and towards the end of the simulation and price drops drastically.

Table 5.10. Evolution of Trading Strategies in Market B

| **Classifier** | **Action** |
|---|---|
| ##1## | 1 |
| ##00# | 3 |
| ###0# | 5 |
| ##### | 3 |

Figure 5.10 provides the volume distribution from the same Market E simulation run.  The volume fluctuates between high volumes followed by lower volume intervals. Higher volume periods correspond to the intervals when traders are making a decision, while lower volume intervals correspond to periods when traders are covering their positions. Volume activity decreases noticeably when the price drops drastically.

The only difference between Market E and Market D is the strategies that agents use, so it is necessary to check the evolution of strategies in Market E to understand what causes the differences in two markets. Table 5.11 shows the most common fired strategies at the end of 1000 runs. Here, the first position in the classifier corresponds to Moving Average (10), the second position represents ROC (5), the third position represents volume, the fourth position represents the conservative biased strategy, and the last position represents trend follower biased strategy. At the end of the simulation, biased strategies survive and dominate the decision process of traders. This can give insight about the drastic price change in Market E. The covering mechanism drives market to calmer periods, where as any disturbance to trader decision making process drives the market to unpredictable dynamics.

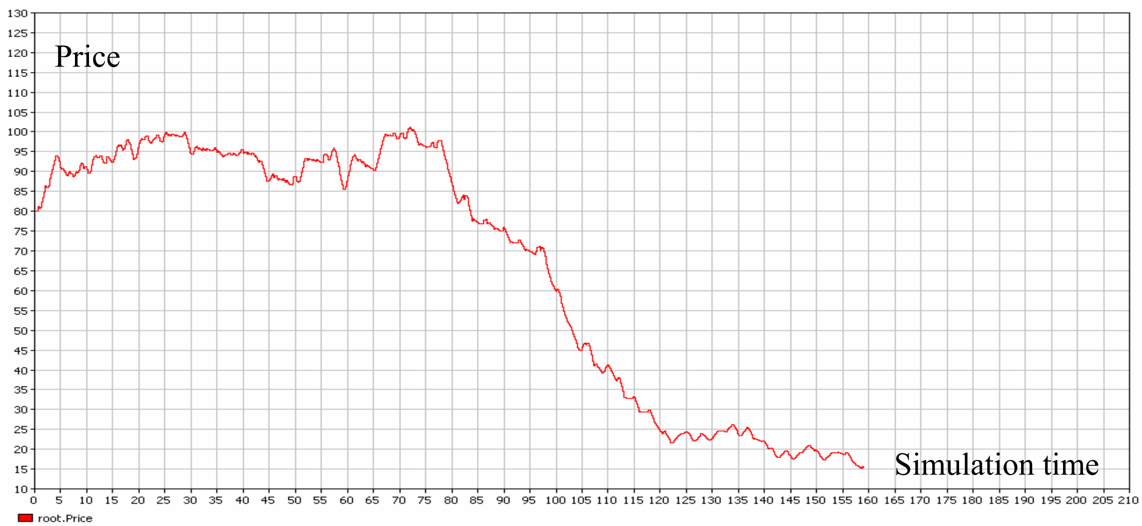


Figure 5.9. Price Formation in Market E

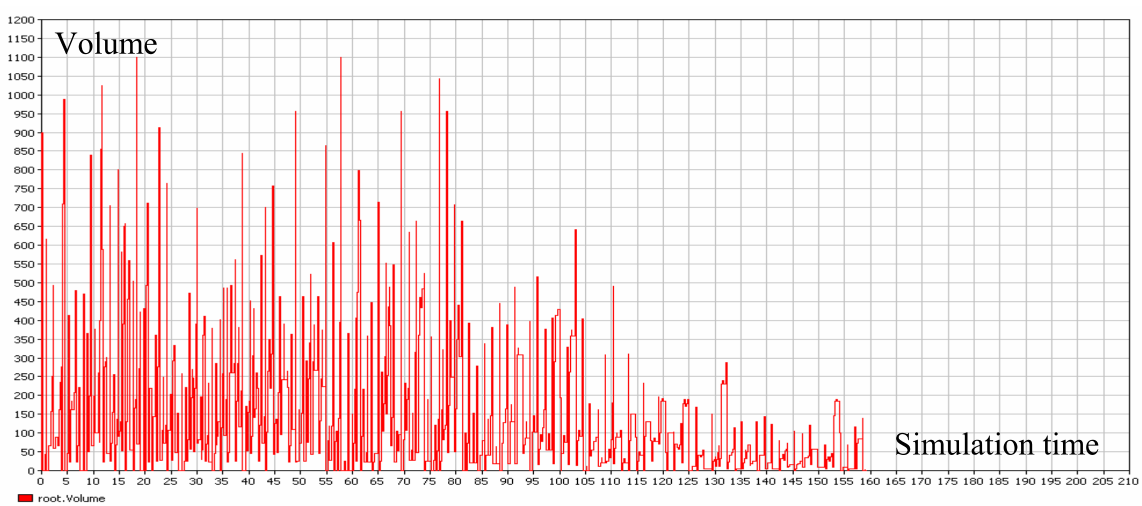


Figure 5.10. Volume Distribution in Market E

Table 5.11. Evolution of Trading Strategies in Market E

| Classifier | Action |
|---|---|
| 0##0# | 1 |
| 1# #0# | 0 |
| ###0# | 3 |
| ###1 | 1 |

**5.5.4. Statistical Properties of the Artificial Time Series.** The statistical properties of the artificial time series provide a degree of validation for the models. It is known that neither the real market time series and nor the stock return series follows a normal distribution. Table 5.12 and Table 5.13 provide the basic statistical properties of the artificial time series and stock return series for Market A, B, C, D and E.

Table 5.12. Statistical Properties of Price Series

| Price Series | mean | $\sigma^2$ | kurtosis | skewness |
|---|---|---|---|---|
| Market A | 100.98 | 11.24 | -1.19 | -0.35 |
| Market B | 88.60 | 6.12 | -0.40 | -0.67 |
| Market C | 79.01 | 3.49 | -0.90 | -0.42 |
| Market D | 76.70 | 1.68 | 0.12 | -0.56 |
| Market E | 66.66 | 31.6 | -1.51 | -0.50 |

Skewness measures the degree of asymmetry in the distribution with the skewness equaling zero in normally distributed series. From the tables, the skewness values show that artificial time series and return series are not normally distributed. Kurtosis provides information about the peaks of the distribution. High kurtosis distribution implies sharper peaks, while low kurtosis implies rounder peaks. Zero kurtosis shows that the series are normally distributed. From the tables, the kurtosis values show that the artificial time series are not normally distributed. The artificial series show characteristics of real-time series to some degree. The artificial price series are asymmetric as real-time series. The artificial series have rounder peaks and show consistent volatility as seen in real markets.

Table 5.13. Statistical Properties of Return Series

| **Return Series** | **mean** | $\sigma^2$ | **kurtosis** | **skewness** |
|---|---|---|---|---|
| Market A | 0.09 | 0.38 | -1.37 | -0.1 |
| Market B | 0.01 | 0.36 | -1.11 | -0.62 |
| Market C | -0.01 | 0.35 | -0.59 | 0.14 |
| Market D | -0.03 | 0.30 | -0.23 | 0.12 |
| Market E | -0.04 | 0.29 | 0.1 | 0.22 |

## 5.6. SUMMARY

This initial study outlines an agent-based financial market simulation derived from the Artificial Life based framework. Five different scenarios are generated to understand the effects of the:

- Covering mechanism on market dynamics
- Learning mechanism on market dynamics
- Biased trading strategies on market dynamics

Since financial markets are self-organized systems where there is no central control, the overall system behavior can be altered by changing the rules of engagement of the environment model. The effect of the covering mechanism on market dynamics illustrates this idea. Market exhibits different behavior when rule of engagement is altered to allow covering mechanism in the market environment. Initial analysis results reveal that when a covering mechanism is incorporated, the market stays in relatively calm periods compared to markets where covering is not allowed. Interestingly, when traders utilize biased trading strategies, the market shows higher fluctuations and more drastic price changes even though a covering mechanism exists. Biased trading strategies are not eliminated under a learning mechanism and play an important role in the aggregate trader behavior.

Evolution is a key characteristic of CAS and SoS. Any framework for analysis of these systems should capture this characteristic. This application study demonstrates this feature through utilization of learning classifier systems. In the market environment, the learning mechanism stabilizes system dynamics and prevents one type of trader from dominating the market. Besides, evolution of trading strategies reveals that strategies based on technical trading perform better under markets where covering is allowed.

The study can provide more information by incorporation of investor psychology models, as well as the addition of rational traders who base their decisions on fundamental stock values. This should show different affects on the outcomes, such as reasons for deviation from rational expectation equilibrium and provide further explanations of the volatility and volume persistence puzzles that traditional models lack in providing insights.

The next section will analyze the financial markets from a different perspective by proposing a trader architecture derived from the Artificial Life framework for SoS analysis. The objective in this study is to incorporate humans as mental processes with error generation mechanism into systems analysis.

# 6. TRADER-BASED ARCHITECTURE AND MARKET BEHAVIOR ANALYSIS

This section extends the artificial stock market model described in Section 5 by proposing a trader architecture, which comprises error generation mechanisms. The objective is to illustrate that humans can be incorporated into systems analysis through different cognitive architectures. This is especially necessary for analyzing the effects of humans on emergent behavior of system architectures. The following sections describe the trader architecture and its effect on the emergent market behavior.

## 6.1. INTRODUCTION

Initial results from Section 5 showed that biased trading strategies survived in a learning environment and that the bias mechanism plays an important role in trader behavior. Therefore, it is necessary to develop a trader architecture that includes human bias mechanisms explicitly. Bias mechanisms model the flaws in information processing and belief formation of humans. Since every behavior starts with perception formation, it is important to understand the effect of this mechanism.

Behavioral finance studies provide models of investor behavior. Behavioral finance is the integration of classical finance with psychology and decision making sciences. It attempts to explain the reasons for some of the anomalies observed in financial markets by studying how investors systematically make errors in judgments (Fuller, 1998). Shefrin (2002) identifies three key themes in behavioral finance:

1. Traders commit errors because they rely on heuristics. Heuristics are used to process data and are generally imperfect. Therefore, traders form biased beliefs that results in irrational decisions. One key issue in behavioral finance is focused on the heuristic-driven biases.

2. Traders' risk and return perceptions are influenced by how decision problems are framed. Another key issue in behavioral finance is focused on the frame dependence preferences.

3. Behavioral finance assumes that heuristic-driven bias and framing effects cause markets prices to deviate from fundamental values. The third key issue in behavioral finance is focused on inefficient markets.

Barberis and Thaler (2003) explain the key issues in behavioral finance as limits to arbitrage and investor psychology:

1. Limits to arbitrage: In the traditional framework, market price equals the fundamental value. This is the discounted sum of expected future cash flows. The Efficient Market Hypothesis states that actual prices reflect fundamental values. While irrational traders are often known as noise traders, rational traders are typically referred to as arbitrageurs. An arbitrage is an investment strategy that offers risk free profits at no cost. Behavioral finance argues that arbitrage strategies designed to correct price deviations from fundamental values can be both risky and costly. Therefore, behavioral finance focuses on the risks and costs associated with arbitrage strategies and why arbitrage can not eliminate deviations from the fundamental value.

2. Investor psychology: Human judgment and decision-making studies contribute to investor psychology studies. Systematic biases arise based on people's beliefs or preferences.

A) Beliefs: How traders form future expectations is an important component of any model of financial markets. Several psychological characteristics that have an impact on expectation formation are listed and briefly described.

- *Overconfidence*: People are overconfident in their judgments. The confidence intervals people assign to their estimates are too narrow.
- *Representativeness*: People often fail to take sample size into account. They infer too quickly on the basis of too few data points and thus find patterns in random events.
- *Conservatism*: If data is not representative of any salient model, people react too little to the data and rely too much on their previous beliefs.
- *Anchoring*: When forming estimates people often start with some initial value, and then adjust away from it. The adjustment is often insufficient, people anchor too much on the initial value.
- *Availability biases (saliency)*: When judging the probability of an event, people often search their memories for relevant information. Recent events and more salient events weigh more and distort the probability estimation.

B) Preferences: Another important part of market models is assumptions about investor preferences, which include how investors evaluate risky options. Most of the models assume that investor preferences are based on the expected utility framework (EU), which is based on maximization of wealth. Experimental work has shown that people systematically violate the EU theory when choosing among risky gambles. Of all the non-EU theories, prospect theory captures experimental results and is suitable for financial applications.

- *Prospect theory:* In Prospect Theory, utility is defined over gains and losses, rather than final wealth positions, as in an expected utility framework. This fits naturally with the way gambles are often presented. It is also consistent with the way people perceive attributes relative to earlier levels rather than in absolute terms. Prospect theory can accommodate the effects of framing. The process by which people formulate problems for themselves is called mental accounting. One feature of mental accounting is narrow framing, which is a tendency to treat individual gambles separately from other portions of wealth (Barberis and Thaler, 2003).
- *Ambiguity aversion:* In reality, probabilities are rarely objectively known. Experiments show that people do not like situations where they are uncertain about the probability distribution of a gamble. Such situations are known as situations of ambiguity. Expected utility does not allow an agent to express their degree of confidence about a probability distribution and therefore can not capture such aversion. Ambiguity aversion is related to how competent an individual feels s/he is at assessing the relevant distribution.

Behavioral finance provides various models that focus on the key issues described above. These models can be incorporated in agent-based models to study the aggregate behavior of traders under adaptive mechanisms. Section 6.2 reviews the bias model which will be incorporated into the artificial financial market.

## 6.2. BIAS MODEL FOR TRADERS

Barberis et al. (1998) propose a bias model for traders. Their model shows two types of bias seen in humans: representativeness and conservatism. These two bias forms play an important role in how humans process new information and form their beliefs. Information processing is an important component in any system, especially in financial markets. Therefore, the focus will be on the effect of these biases on market dynamics.

**6.2.1. The Biased Trader Model.** Summarizing Barberis's model (Barberis et al. 1998), stock earnings are given as dividends and dividends follow a random walk. The model's main assumption is that investors believe dividends move between two Markov states (Barberis et al. 1998). A change in earning period (t) depends only on the change in earning period (t-1). In the first state, a positive earning shock is likely to be reversed to a negative shock in the following period with some transition probability, and vise versa. In the second state, a positive earning shock is likely to be followed by another positive shock with some transition probability, or a negative shock is likely to be followed by another negative shock. State 1 models conservative behavior of humans and state 2 models the trend following representative behavior of humans. The transition probabilities in both states are fixed in the trader's mind. Each period, when the dividend is revealed, the trader uses this information to update beliefs about which state the market is in. There is also a model switching process that determines which state to use. This is also a Markov process where the transition probability of switching from State 1 to State 2 is low, so State 1 dominates the bias model.

Tables 6.1 and 6.2 summarize the bias mechanism in two state models. Here, y(t) is the dividend change in period t, and y(t+1) is the dividend change in period t+1. *P* is the probability that positive shock is likely to be reversed and *pr* is the probability that a positive shock is likely to be followed. Table 6.3 shows the Markov process that determines which state Model a trader uses to make a decision. Table 6.4 provides the transition probability value ranges for all three Markov processes.

Table 6.1. Model 1-Conservatism

| **Model 1** | y(t+1)=y | y(t+1)=-y |
|---|---|---|
| y(t)=y | P | 1-P |
| y(t)=-y | 1-P | P |

Table 6.2. Model 2- Representative Trend Following

| **Model 2** | y(t+1)=y | y(t+1)=-y |
|---|---|---|
| y(t)=y | pr | 1-pr |
| y(t)=-y | 1-pr | pr |

Table 6.3. Transition Probabilities from Model 1 to Model 2

| | (t+1)=M1 | (t+1)=M2 |
|---|---|---|
| y(t)=M1 | 1-x | x |
| y(t)=-M2 | z | 1-z |

Table 6.4. Transition Probability Values

| | |
|---|---|
| P | 0-0.5 |
| pr | 0.5-1 |
| x and z | <0.5 |

These probability values produce time series that are consistent with available statistical evidence. They are also consistent with experimental evidence on the failures of individual judgment under uncertainty and the trading patterns of investors in experimental settings. Therefore, similar probability values will be used in our study. Their bias model is developed for a representative agent, so incorporating this model to an agent-based framework can provide more insights on the aggregate effect of biased behavior on market dynamics.

**6.2.2. Justification for the Biased Trader Behavior.** Statistical evidence from analysis of historical time series provides evidence of the conservative and trend follower behavior of traders. Also, psychological experimental studies on failures of individual judgment under uncertainty and the trading patterns of investors under controlled stock trading experiments provide additional justification for the assumptions of biased trader behavior models.

Barberis et al. (1998) summarize several experimental studies of psychologists related to these phenomena. In one experimental study (Barberis et al., 1998), a trader's reaction to new evidence is benchmarked against rational Bayesian belief update. The results showed that traders update their beliefs slow in magnitude compared to the rational model. It took two to five observations more to change and update their beliefs compared to the rational model.

Conservative behavior is closely related to the under-reaction behavior seen in real markets. Under-reaction to news announcements means that stock returns following good news announcement are greater than stock returns following bad news. Stock under reacts to good news and corrects it in the next period by giving higher returns. There are several empirical studies of historical real time series that provide evidence for this type of behavior. One study examines the cross-section of U.S. stock returns and finds that stock returns are positively auto-correlated over a six-month horizon (Barberis and Thaler, 2003) and interpret this to slow incorporation of information into prices. Another statistical study calculates the standardized unexpected earnings (SUE), which is a scaled value of the difference between company's current earnings and its earnings one year earlier. The results show that stocks with highest SEU earn 4.2% higher returns than stocks with lowest SEU, which is further evidence of under reaction (Bernard, 1992).

The other psychological evidence is the representative heuristic. This behavior is closely related to over-reaction seen in real markets. Over-reaction occurs especially when a company has consistent earnings growth over several years. Investors may think that past performance history is representative of future earnings, while the historical earnings growth can be random. As a result, investors get disappointed when future earnings fail to support their expectations. There are several empirical studies that provide evidence of this type of behavior. Analysis of cross-section of stock returns reveal that stocks that had low returns over the previous five years outperformed stocks that had high returns for the same time period (Barberis and Thaler 2003, Shleifer, 2000). This suggests that stocks with historically high returns are over valued and stocks with historically low returns are under valued. Therefore, traders can earn returns by betting against representative behavior. Another study by La Porta (1996) sorts stocks based on growth rate forecasts made by professional analysts and finds that analysts are bullish about the stocks they are optimistic on and bearish about stocks that are pessimistic about. Therefore, representative behavior can be seen also among professional analysts.

Besides empirical studies, Griffin and Tversky (1992) develop a framework that combines conservative and representative behaviors. In their model, people update beliefs based on strength and weight of new information. Strength means the extremity of the information. Weight means the statistical sample size of the information. In their framework, Griffin and Tversky claim that people pay too much attention to strength of the information and too little on the weight of the information. Therefore, when information has low strength and high weight, people under react and conservative behavior occurs. When information has high strength but low weight, people over react and representative behavior occurs (Scheifer, 2000).

Based on these statistical, experimental and theoretical studies, Barberis et al. (1998) develop the Markov based trader belief formation model. They find transition probability ranges that exhibit under reaction and over reaction behavior observed in empirical studies.

### 6.3. TRADER-BASED ARCHITECTURE

The trader-based architecture combines the Markov based bias model and XCS trader learning mechanism. Traders still have technical trading strategies that evolve based on market dynamics, but they also have a bias model that interrupts their learning mechanism. Traders mainly use the XCS mechanism to update their trading strategies and make a decision, but at some intervals with some probability they switch to biased models to make a decision. This switching mechanism to bias model is in a way analogous to how some information shock from the environment can lead to different perceptions and behavior. Figure 6.1 summarizes the cognitive architecture of the trader.

In terms of Sloman's H-Cogaff architecture, a meta-management module determines which reasoning mechanism will determine the behavior of the trader. This is also a Markov process. A reactive mechanism of the architecture is not necessary for the market environment because traders make decisions based on deliberative reasoning. A reactive module is vital in environments where immediate response is necessary. Figure 6.2 explains trader architecture in terms of H-Cogaff architecture (Kilicay et al., 2007b).

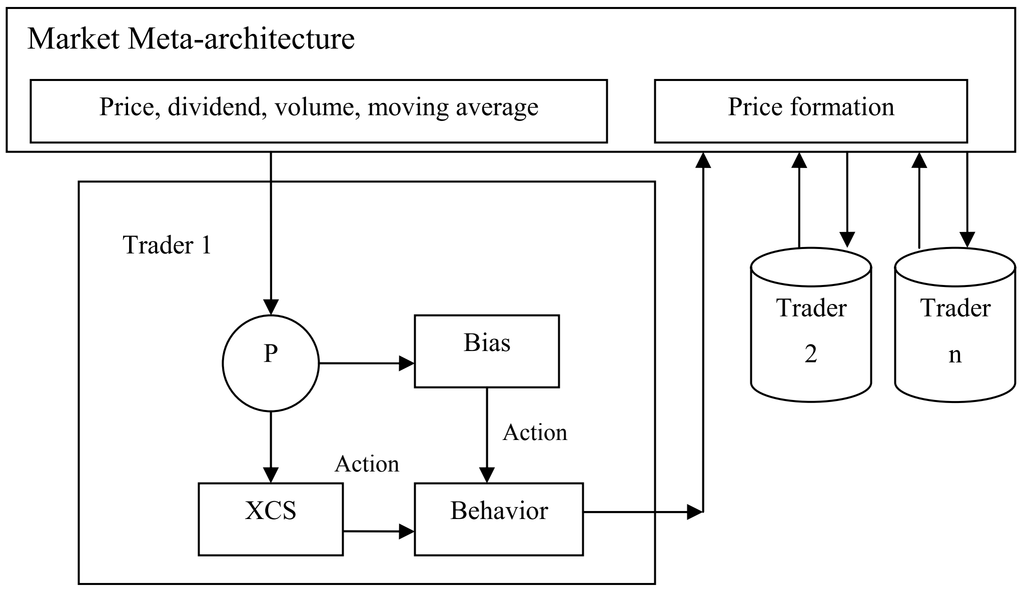


Figure 6.1. The Trader-based Architecture

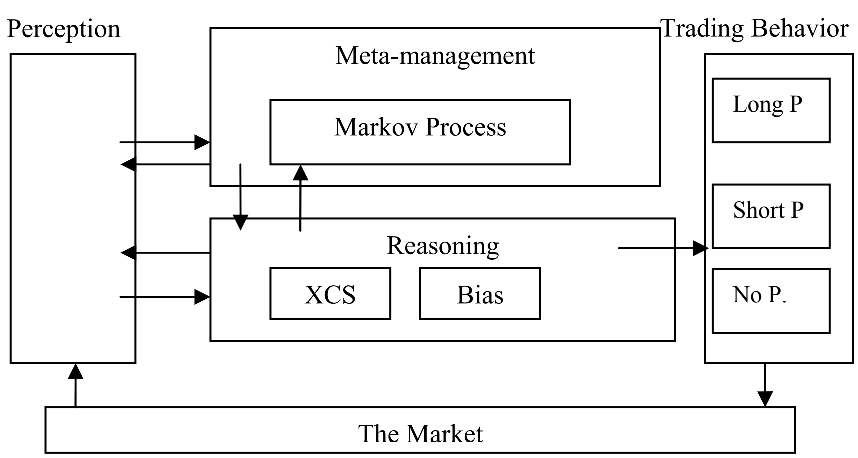


Figure 6.2. H-Cogaff Architecture Applied to Trader Architecture

## 6.4. SIMULATION MODEL

The model is similar to the artificial stock market model described in Section 5. The agent types, price formation mechanism, the evolution mechanism and agent behavior are the same. There is one adjustment to agent's trading rules, which is the addition of fundamental trading rule to agent's decision rules. Each trader also has the option of calculating the intrinsic value of the stock using a dividend discount model and then comparing whether the fundamental value is greater than the stock price in the market. If fundamental value is greater than the market value, the trader takes a long position, else a short position is taken.

The dividend discount model is a stock valuation tool that calculates the present value of the future dividends that a company is expected to pay to its shareholders. It allows investors to determine the intrinsic value of a stock that is not influenced by current stock market conditions. Equation 9 describes this model:

$$P_{fun} = \frac{Div_1}{(1+r)} + \frac{Div_2}{(1+r)^2} + \frac{Div_3}{(1+r)^3} + \ldots = \frac{Div}{r} \tag{9}$$

where $P_{fun}$ is the fundamental price of the stock, Div is the dividend and r is the rate of return. The trader classifier contains this additional information in the new artificial stock market.

Initially, the dividend is revealed before the stock price is determined. The dividend is assumed to follow the auto-regressive process. Equation 10 shows the model used for dividend process.

$$d_t = d_{mean} + \rho(d_{t-1} - d_{mean}) + \varepsilon \tag{10}$$

where $d_t$ is dividend at time t, $d_{mean}$ is dividend mean, $\rho$ = 0.95 and $\varepsilon$ is the error value that exhibits N (0, $\sigma^2$ ). This process provides persistence in the dividend process without getting close to a non-stationary dividend processes (LeBaron, 1999).

The simulation starts with an initial population of long traders, short traders and traders who do not take any position. This is necessary to create initial price series.

Traders, based on their behavior type, determine their demand/supply for the shares. Market price is determined from the aggregate demand/supply of traders. Once price is revealed, the market calculates and stores several technical indicators, such as moving averages and trading volume. The fundamental value of the stock is also calculated and revealed at this time. In the next trading period, the meta-management layer of the trader architecture determines the reasoning mechanism that will determine the behavior. At this point the meta-management determines the reasoning mechanism based on a pre-determined probability value. If the XCS mechanism is selected, the trader gets the current market technical indicators, as well as the fundamental value of the stock. The system selects an action from trader's trading rules and this action fires a specific trader behavior. The system rewards trading rules if the trader is successful in covering their position. If the bias mechanism is selected as the reasoning mechanism, the trader gets the current dividend value and determines the current state of the market based on the Markov process described in Section 6.2.2. Figure 6.3 shows the simulation architecture and the relationship between the sub-components and the system level.

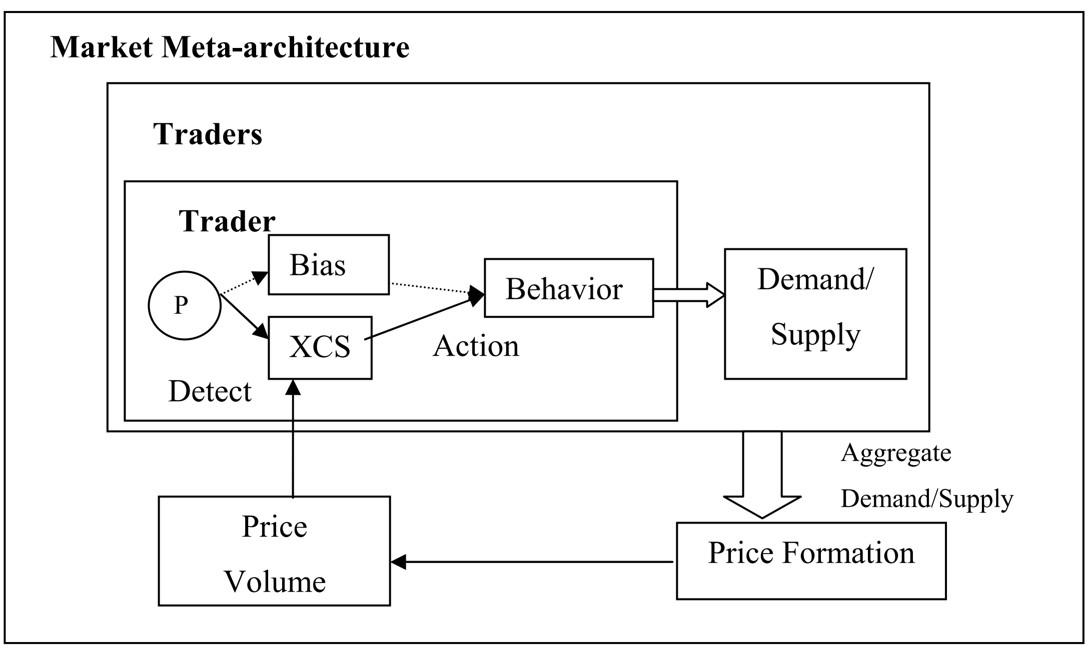


Figure 6.3. Simulation Architecture

**6.4.1. Sensitivity Analysis.** When the outputs of computational models are time series or functions of other continuous variables, the main interest is in the general pattern or structure of the curve. In these cases, model sensitivity focuses on the effect of model input choices on the overall shapes of output curves (Williams et al., 2005). The artificial stock market model output is a time series and thus model sensitivity focuses on the effect of model inputs on the price series curve characteristics. It is known that neither the real market time series nor the stock return series follows a normal distribution. Skewness measures the degree of asymmetry in the distribution and skewness equals zero in normally distributed series. Therefore, in analyzing the model, the interest is on what shifts the curve up or down, moves it left or right, or what makes the peaks wider or narrower. The sensitivity analysis conducted for the artificial stock model focuses on these two characteristics of the price series. Input parameters are changed to observe whether they have a significant effect on the skewness, kurtosis, mean and standard deviation characteristics of the price series. The parameter ranges are selected so that the price series curves are not normally distributed. Table 6.5 provides the input parameters that were analyzed for the sensitivity analysis for the artificial stock market model.

Table 6.5. Model Input Parameters Analyzed for Sensitivity Analysis

| **Parameter** | **Description** | **Tested parameter range** |
|---|---|---|
| $r_f$ | Risk free rate | 0.01-0.1 |
| $r_a$ | Risk aversion | 0.5-1.5 |
| v | Volatility | 0.1-0.6 |
| N | Total number of shares | 1000-2000 |

The parameters in Table 6.5 have a small influence on the outcome of the simulation. A variation of their values within a reasonable range has a small affect on the overall results of the simulations. Two parameters have a bigger influence on the outcomes of the simulation. There is a need to search for appropriate values for these parameters so that the artificial price series exhibits empirical stylized facts. These two parameters and the tested range for these parameters are listed in Table 6.6. Figures 6.4 through 6.9 provide the price series generated by changing these two parameters.

Table 6.6. Model Input Parameters that have Larger Effect on Model Outcomes

| **Parameter** | **Description** | **Tested parameter range** |
|---|---|---|
| alpha | Price adjustment rate | 0.0001-0.001 |
| a | Expected price change percentage | [0.002-0.1], [(-0.005) - (-0.1)] |

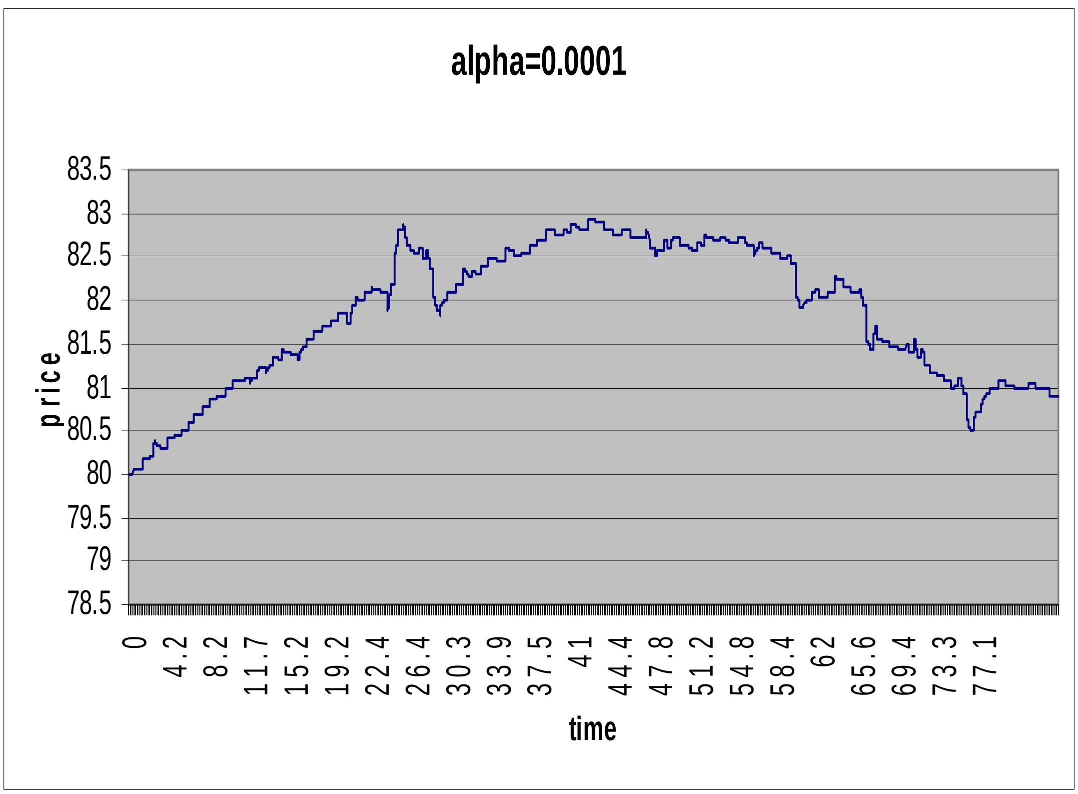


Figure 6.4. Price Series Alpha=0.0001

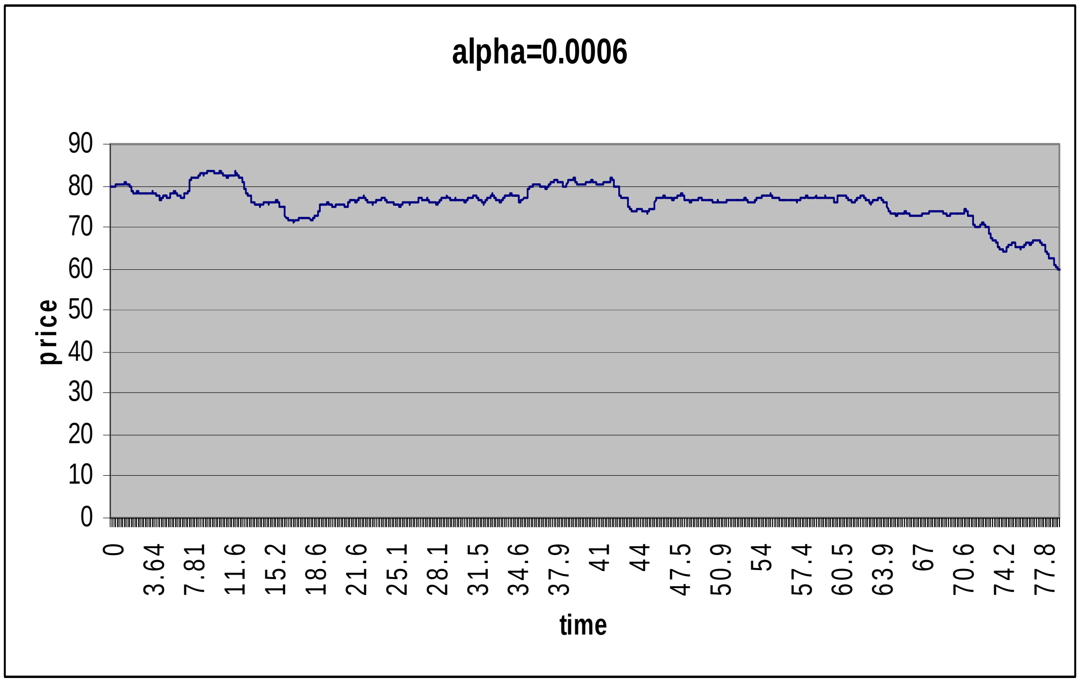


Figure 6.5. Price Series Alpha=0.0006

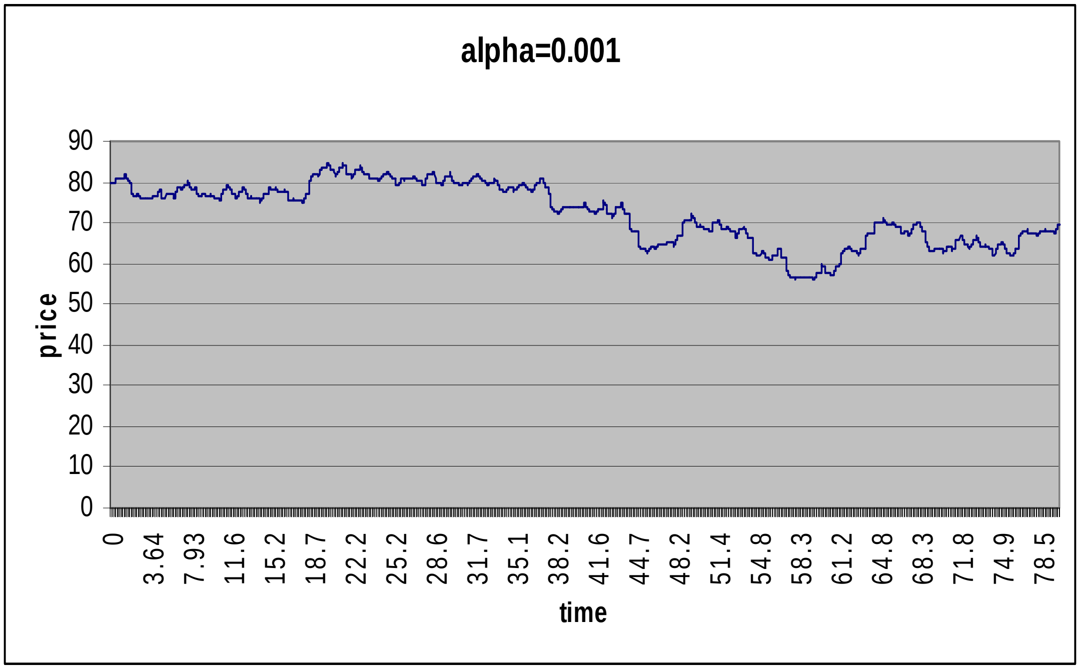


Figure 6.6. Price Series Alpha=0.001

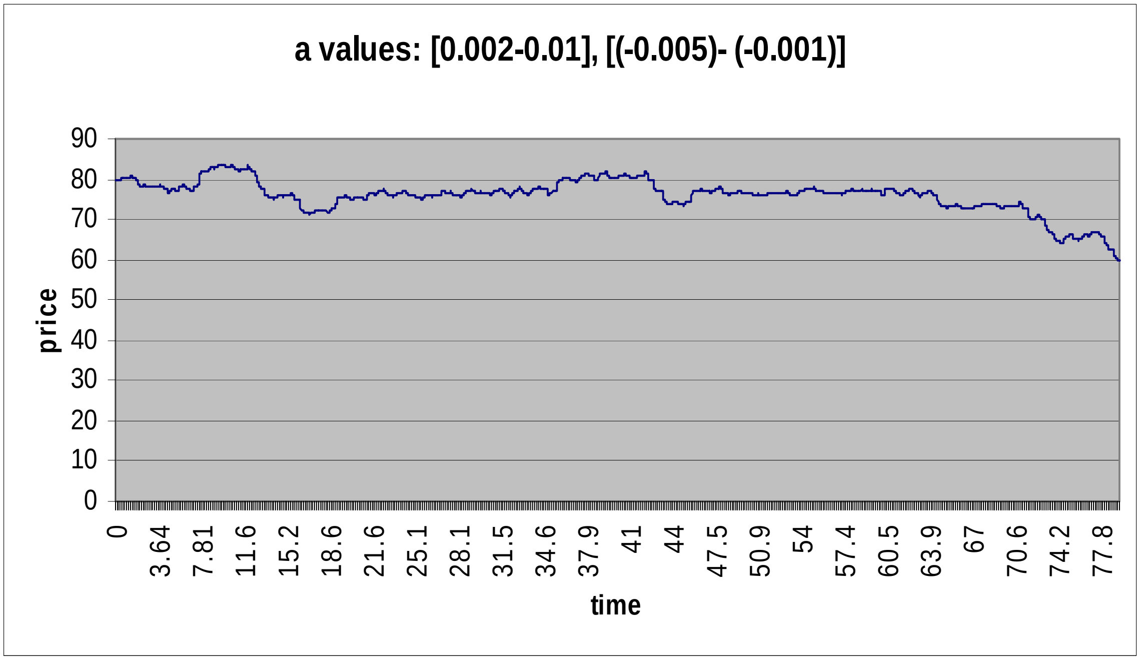


Figure 6.7. Price Series A= [0.002-0.01], [(-0.005)-(-0.001)]

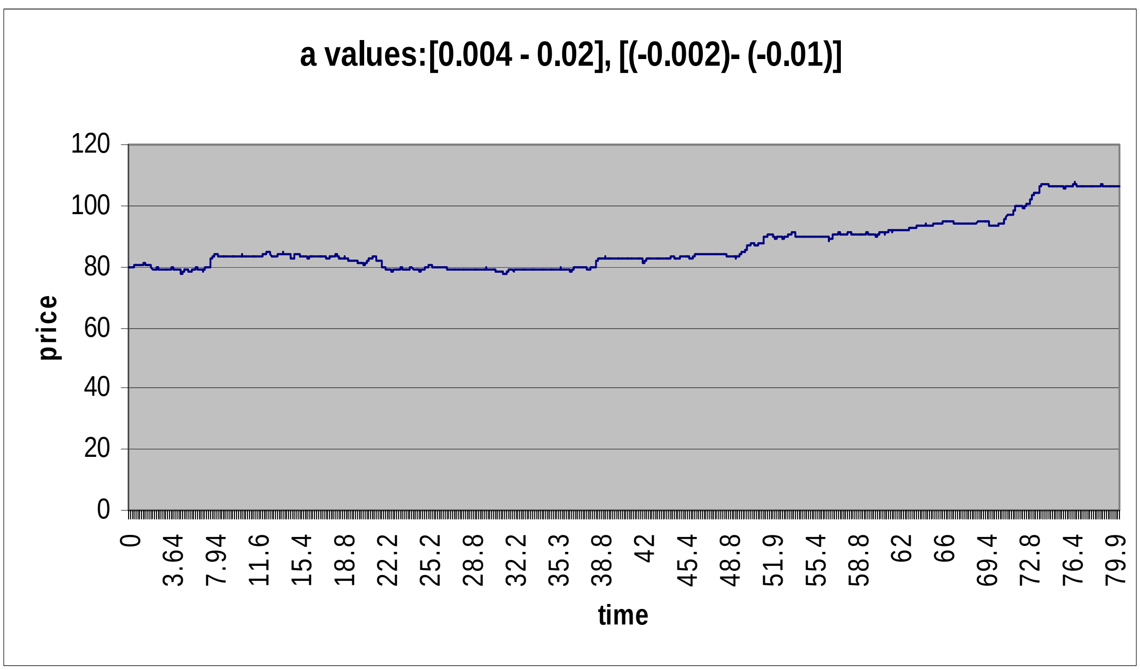


Figure 6.8. Price Series A= [0.004-0.02], [(-0.002)-(-0.01)]

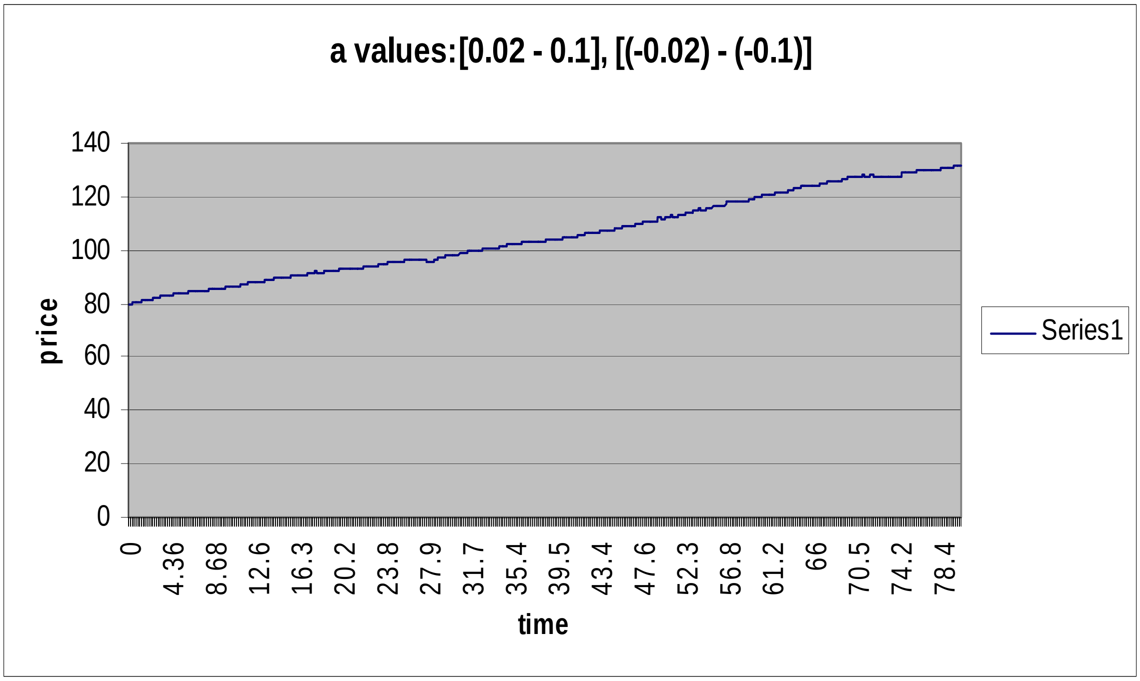


Figure 6.9. Price Series A= [0.02-0.1], [(-0.02)-(-0.1)]

Table 6.7 provides the skewness, kurtosis, mean and standard deviation values of the price series generated from sensitivity analysis for parameter alpha. When alpha value is small (alpha=0.0001), the standard deviation and skewness of the price curve moves towards normally distributed curve characteristics. When the alpha value is increased (alpha=0.0006), the curve characteristics follow the characteristics of real price series, including positive kurtosis, which confirms the fat tail phenomenon observed in real market time series. When the alpha value is increased more (alpha=0.001), the standard deviation of the artificial price series increases drastically, but the fat tail phenomenon is not seen in this case.

Table 6.8 provides the skewness, kurtosis, mean and standard deviation values of the price series generated from sensitivity analysis for parameter (a), the expected price increase percentage value. When the parameter value is selected between (0.002,-0.005), the artificial time series exhibit the characteristics of real time series including, the fat tail phenomenon. When the parameter values are increased between (0.004, - 0.02), the price series characteristics are still not normally distributed, but the fat tail phenomenon is not seen. When the parameter values are increased between (0.02, -0.1), the price series start to shift towards normal distribution characteristics.

Table 6.7. Statistical Analysis of the Price Curve when Alpha Parameter is Changed

| | **Kurtosis** | **Skewness** | **Mean** | **Standard Deviation** |
|---|---|---|---|---|
| Alpha=0.0001 | -1.11 | -0.37 | 81.83 | 0.78 |
| Alpha=0.0006 | 2.28 | -1.31 | 75.69 | 4.54 |
| Alpha=0.001 | 0.66 | 1.26 | 81.87 | 19.32 |

Table 6.8. Statistical Analysis of the Price Curve when the Parameter *a* is Changed

| | **Kurtosis** | **Skewness** | **Mean** | **Standard Deviation** |
|---|---|---|---|---|
| *a*= [0.002, -0.005] | 2.28 | -1.31 | 75.69 | 4.54 |
| *a*= [0.004, -0.02] | -1.11 | 0.64 | 90.21 | 10.86 |
| *a*=[0.02, -0.1] | -1.31 | -0.08 | 113 | 18.54 |

**6.4.2. Model Parameters.** The model parameters are selected after the sensitivity analysis. The parameters that the model output is sensitive (alpha and *a*) are adjusted so that the price series exhibit real market price series characteristics. Other parameters that the model is not significantly sensitive to are selected from the values that are used in similar artificial financial market studies. For example, the risk aversion is selected from (Takashi, 2002), the total number of stocks is selected from (LeBaron, 1999), and the number of traders is selected from (Pfister, 2003). The risk free rate is selected from the last quarter value of 2005. Table 6.9 provides the parameter values utilized in the simulation experiments.

Table 6.9. Simulation Parameters

| **Parameter** | **Description** | **Value** |
|---|---|---|
| N | Number of traders | 100 |
| S | Total number of stocks | 1100 |
| $\lambda$ | Risk aversion | 1.25 |
| $r_f$ | Risk-free interest rate | 0.04 |
| $\sigma^2$ | volatility | 0.12 |
| T | Time for covering | Uniform(0.2-0.5) |
| alpha | Price adjustment factor | 0.0006 |
| $a$ | Expected price increase/decrease percentage | (0.002, -0.005) |

**6.5. THE BENCHMARK MODEL**

For artificial financial markets, a benchmark is useful for comparing the simulation results. LeBaron (2001) utilizes the homogenous agent environment as the appropriate benchmark for multi-agent simulations. For the artificial stock market, homogenous rational expectations equilibrium is utilized as the benchmark for comparison of results.

Under Constant Absolute Risk Aversion (CARA) utility, the demand for the risky asset is given as Equation 7 in Section 5.4.4. This demand equation holds true in the linear rational expectations equilibrium.

The traders assume that the price of the risky asset is a linear function of the dividend. Equation 11 gives this linear assumption.

$$p_t = fd_t + e \tag{11}$$

where $p_t$ is the current price, $d_t$ is the current dividend, $f$ and $e$ are the linear relationship coefficients. Equation 11 can be further extended by utilizing the dividend process from Equation 10 in Section 6.5. Since the agents in the benchmark model are identical with the same coefficient of absolute risk aversion ($\lambda$), homogeneous linear rational expectations equilibrium price ($P^*$) can be calculated by incorporating the linear price equation into the demand equation given in Equation 7 in Section 5.4.4. Forcing each trader to optimally hold one share at all times and solving for f and e coefficients gives Equations 12 and 13, respectively.

$$f = \frac{\rho}{1 + r_f - \rho} \tag{12}$$

where $\rho$ is a constant derived from dividend process in Equation 10 in Section 6.5, $r_f$ is the risk free interest rate.

$$e = \frac{d_{mean}(f+1)(1-\rho) - \lambda \sigma_d^{\,2}}{r_f} \tag{13}$$

where $d_{mean}$ is the dividend mean, $\lambda$ is the risk aversion constant, $\sigma_d^{\,2}$ is the dividend error variance.

Therefore, the rational expectations equilibrium price ($P^*$) can be given as shown in Equation 14.

$$P^* = \frac{\rho}{1 + r_f - \rho} d_t + \frac{d_{mean}(f+1)(1-\rho) - \lambda \sigma_d^{\,2}}{r_f} \tag{14}$$

Table 6.10 provides the parameter values used in the simulation to calculate the rational expectations equilibrium price. The risk aversion constant, risk free rate are the same as the market simulation parameter values. The dividend process constant and dividend mean are selected from similar artificial stock market studies (Pfister, 2003 and LeBaron, 2001). The dividend error variance and the linear relationship constants are calculated to capture the rational expectations equilibrium dynamics.

Table 6.10. Rational Equilibrium Price Parameter Values

| **Parameter** | **Simulation value** |
|---|---|
| $\gamma$ | 1.25 |
| $d_{mean}$ | 3.0 |
| $r_f$ | 0.04 |
| $\rho$ | 0.95 |
| $\sigma^2$ | 0.07 |
| $f$ | 10.55 |
| $e$ | 41.13 |

## 6.6. EXPERIMENTS AND RESULTS

Simulation experiments are generated using the AnyLogic5.1 simulation software™. Five replications of the same simulation are conducted to observe if the results are consistent between replications. Each simulation takes 1000 trading periods. The replications revealed the same results. In order to understand the effect of bias mechanisms, three scenarios are considered. In scenario 1, there is no bias mechanism. Traders mainly utilize the learning mechanism to make decision. In scenario 2, the bias mechanism is included and the probability of trader using the bias mechanism is set to 0.4. In scenario 3, the probability of trader using the bias mechanism is set to 0.8. The learner and biased traders are tracked in the simulation. In all scenarios, covering is allowed.

Based on these different scenarios, simulations are conducted using the parameters shown in Tables 6.9 and 6.10. The time series graphs in the next section provide information about the effect of bias mechanisms on the overall market dynamics.

**6.6.1. Experiments.** Three different probability values are tested for switching from learning mechanism to bias mechanism. This provides insights to market behavior dynamics. The effect of bias mechanism on learning mechanism is determined by analyzing the performance of the XCS under different trader probability values.

In Scenario 1, traders only utilize the learning mechanism. Therefore, the probability of trader using the bias mechanism is set to zero. Figure 6.10 illustrates the price formation in Scenario 1. The price fluctuates around 75-85. The price movement shows relatively calmer market dynamics. This is due to the covering mechanism. This characteristic was highlighted in the experimental results in Section 5. Figure 6.11 provides the volume distribution from the same simulation run. Higher volume is followed by lower volume periods.

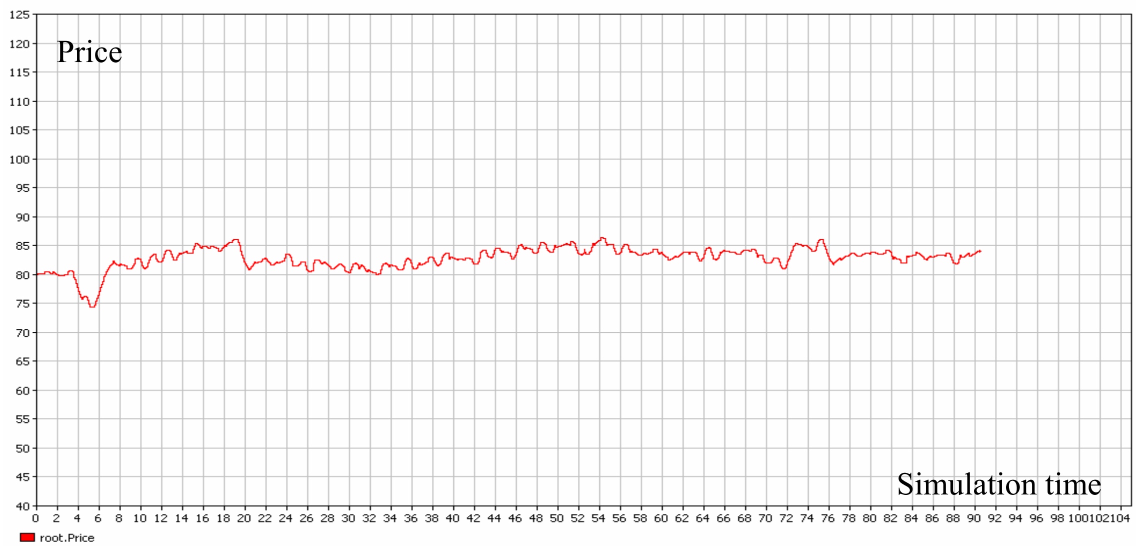


Figure 6.10. Price Formation in Scenario 1

In Scenario 2, the probability of trader using the bias mechanism is set to 40%. Figure 6.12 shows the price formation when there is a significant amount of biased traders in the market. The price fluctuates around 80-105 dollars, showing an upward trend. Figure 6.13 shows the volume distribution from the same simulation run. Higher volumes indicate that traders are taking a position, whereas lower volumes indicate traders covering their position. When traders are covering their position, they are making a trade, but because each trader's covering time is different it reflects as lower volume.

Figure 6.14 and Figure 6.15 show the distribution of traders utilizing the learning mechanism and bias mechanism to make a decision. In Scenario 2, the learners and bias traders are almost equal in amount.

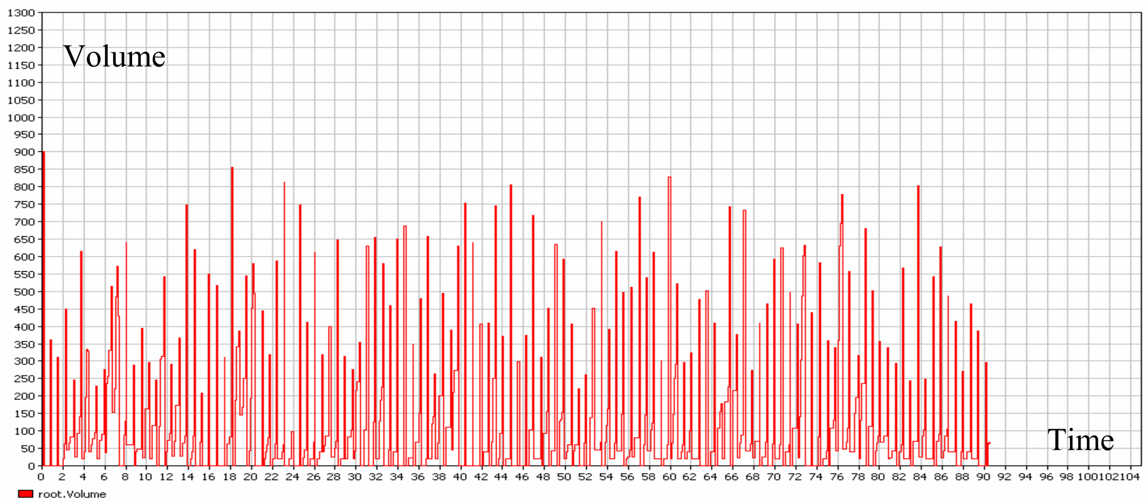


Figure 6.11. Volume Distribution in Scenario 1

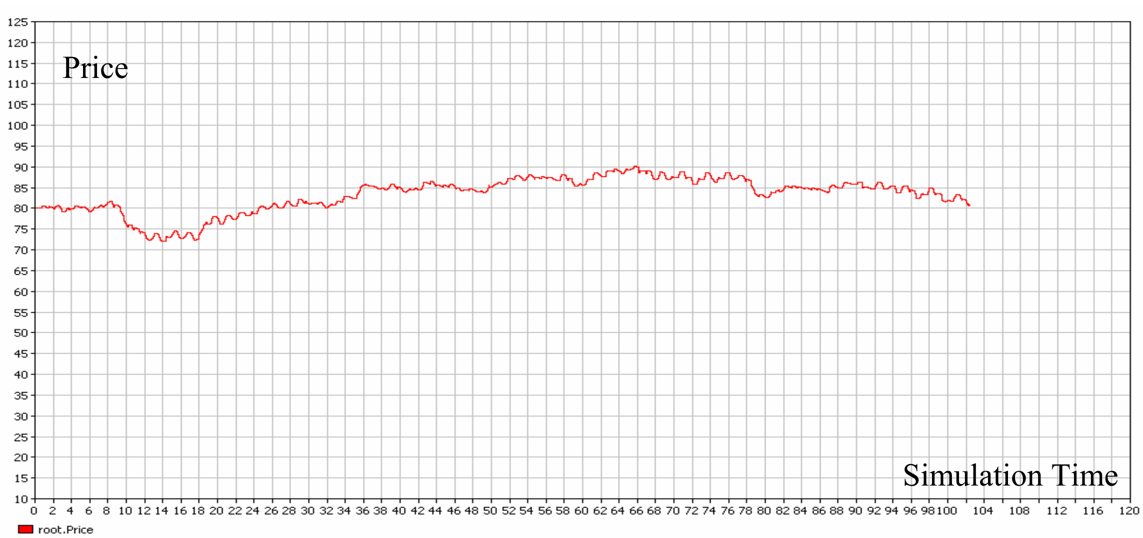


Figure 6.12. Price Formation in Scenario 2

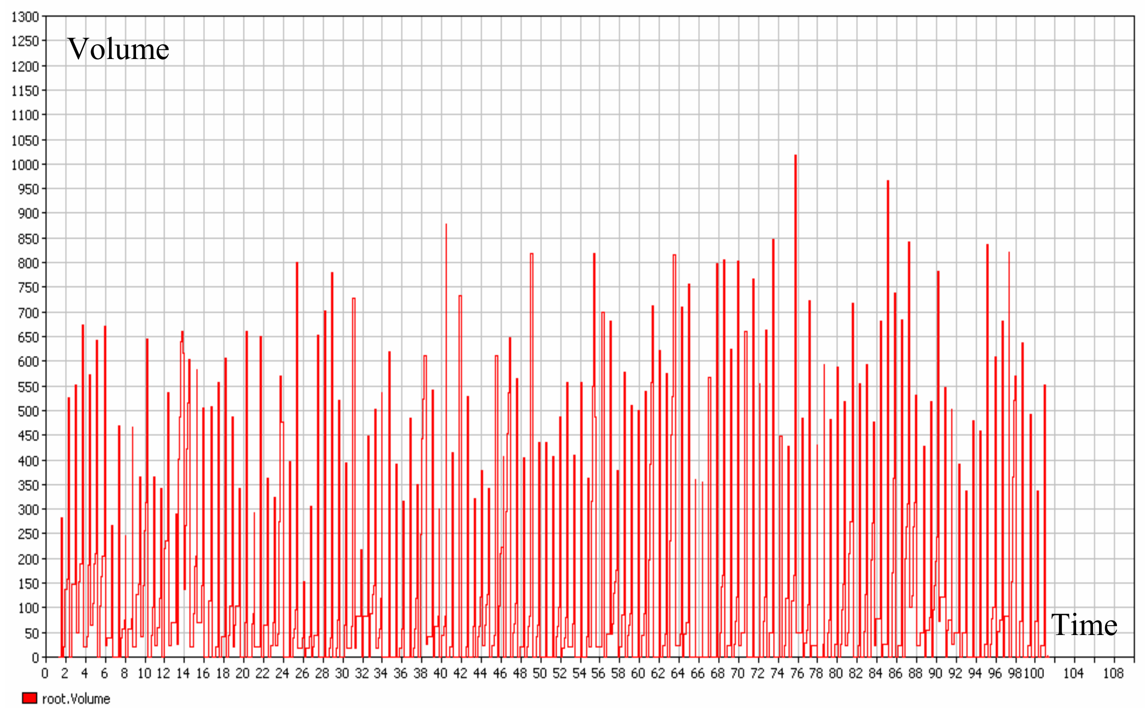


Figure 6.13. Volume Distribution in Scenario 2

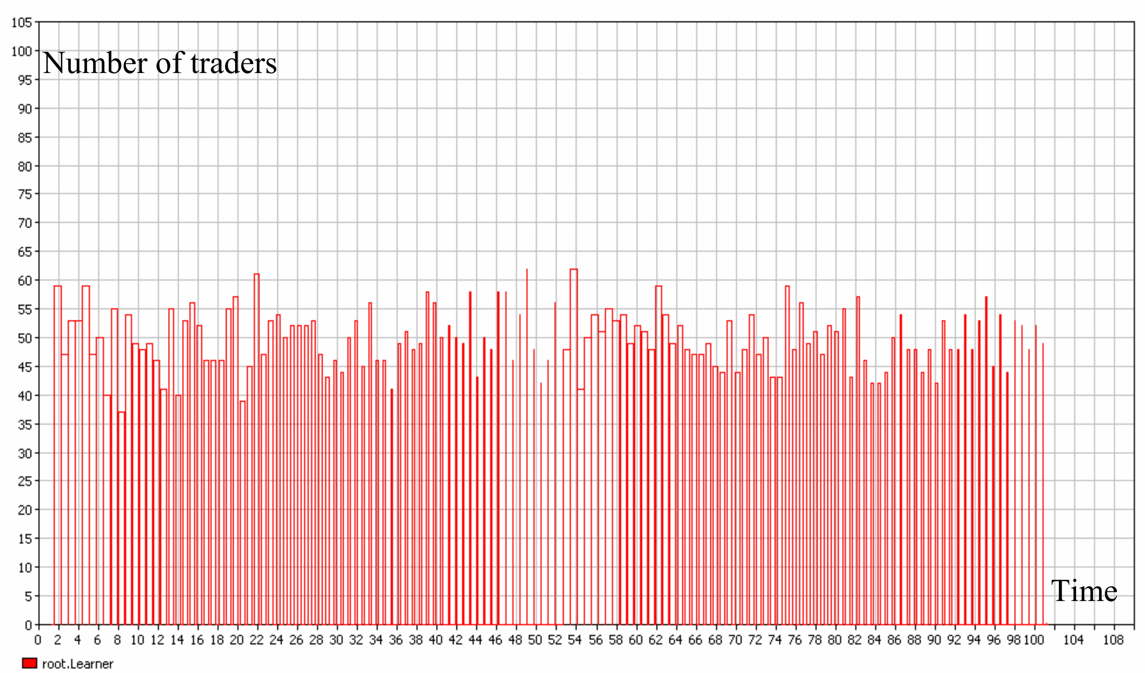


Figure 6.14. Distribution of Traders Using the Learning Mechanism in Scenario 2

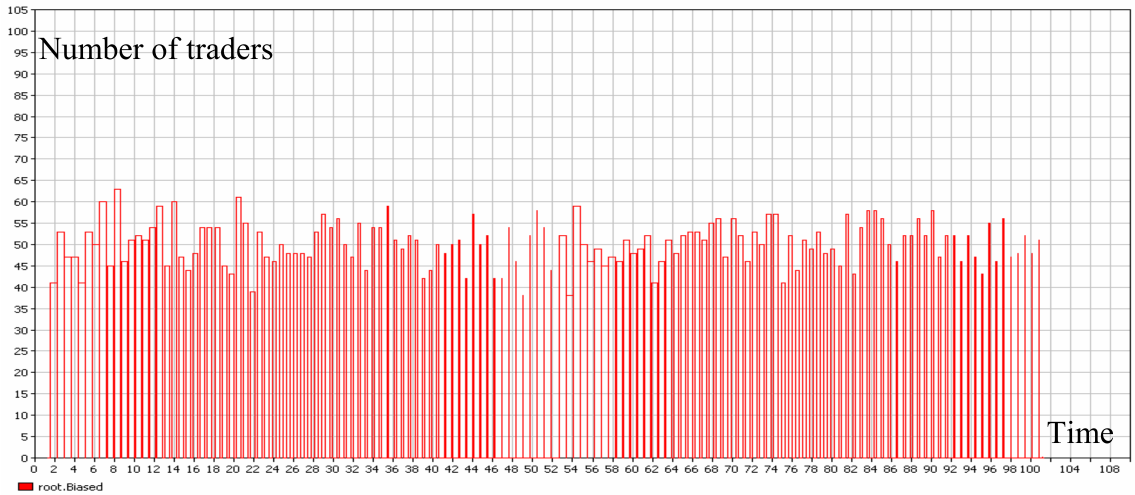


Figure 6.15. Distribution of Traders Using the Bias Mechanism in Scenario 2

In Scenario 3, the probability of a trader using bias mechanism is set to 80%. Figure 6.16 shows the price formation when the bias mechanism dominates trader's behavior. The price fluctuates between 70-90 dollars. Figure 6.17 shows the volume distribution in Scenario 3. Figure 6.18 and Figure 6.19 show the distribution of traders utilizing the learning mechanism and bias mechanism to make a decision. In Scenario 3, the bias traders dominate the market.

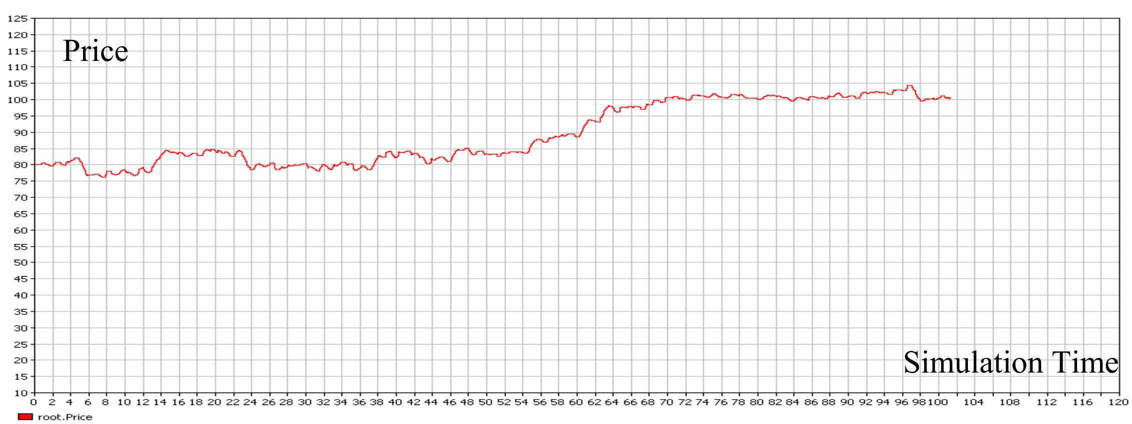


Figure 6.16. Price Formation in Scenario 3

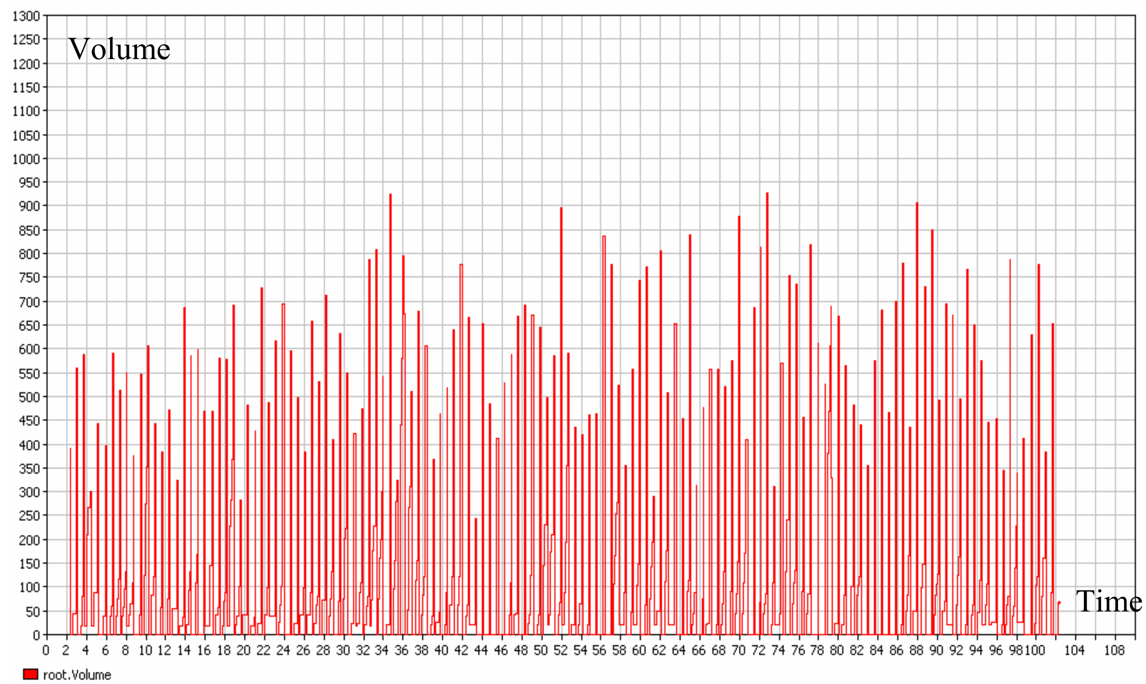


Figure 6.17. Volume Distribution in Scenario 3

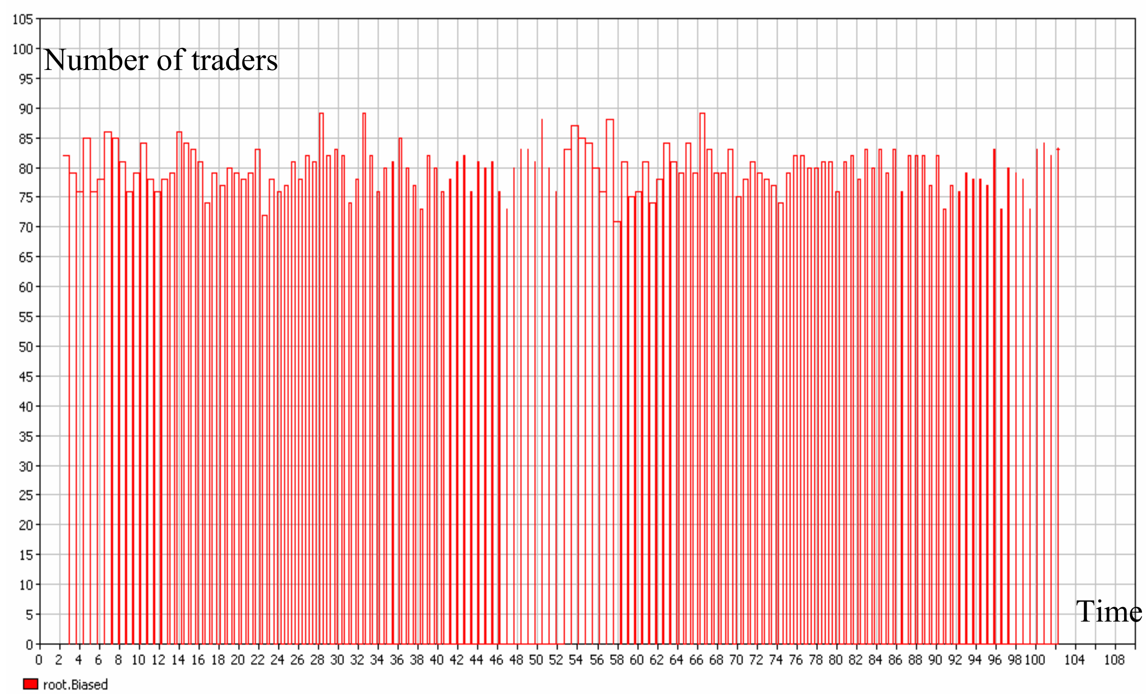


Figure 6.18. Distribution of Traders Using the Bias Mechanism in Scenario 3

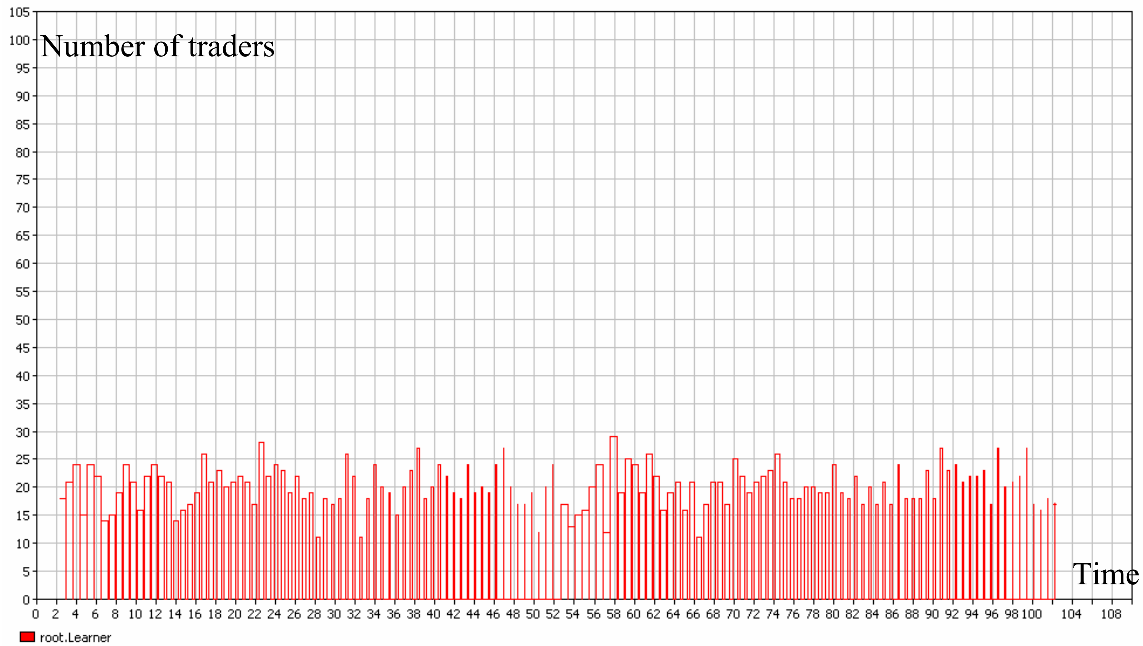


Figure 6.19. Distribution of Traders Using the Learning Mechanism in Scenario 3

The effect of bias mechanism on learning mechanism:

Since the trader switches back and forth between the learning mechanism and the bias mechanism, the bias mechanism affects the performance of the learning mechanism. Figure 6.20 illustrates the system performance over 1000 iterations when the rule population size is 800 and there is no bias mechanism disturbing the learning mechanism (Scenario 1). System performance for XCS learning classifier system is the fraction of the last 50 exploit trials that were correct (Wilson, 1995). Figure 6.21 illustrates the system performance over 1000 iterations when there is a 40% probability of switching to bias mechanism (Scenario 2). From the graphs, it can be seen that the system performance is in a continuous upward trend when there is no bias mechanism. However, when there is bias mechanism, the system performance for the learning mechanism increases slower. The 1000 iterations are not enough for the hybrid system (Scenario 2) to reach the same performance level as the Scenario 1.

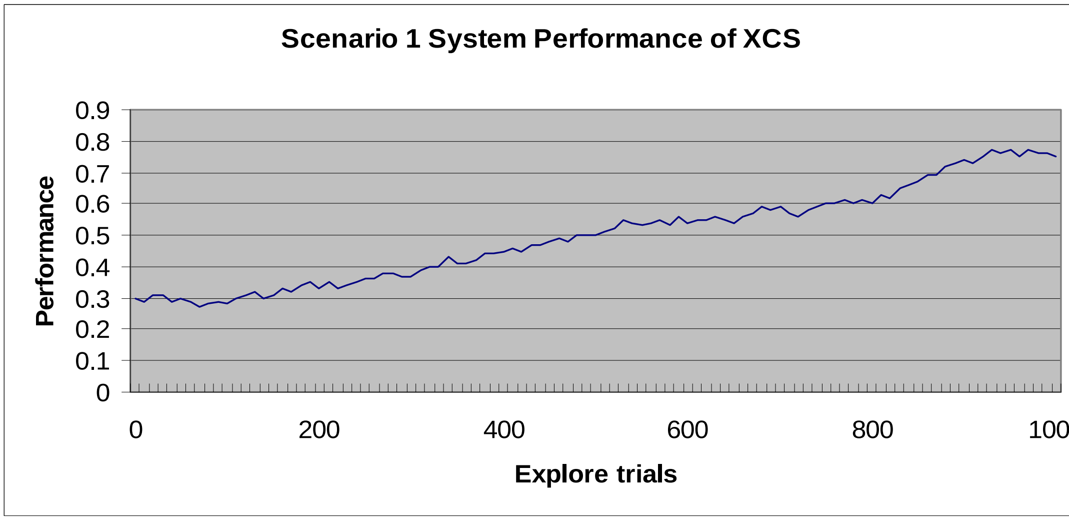


Figure 6.20. System Performance of XCS in Scenario 1

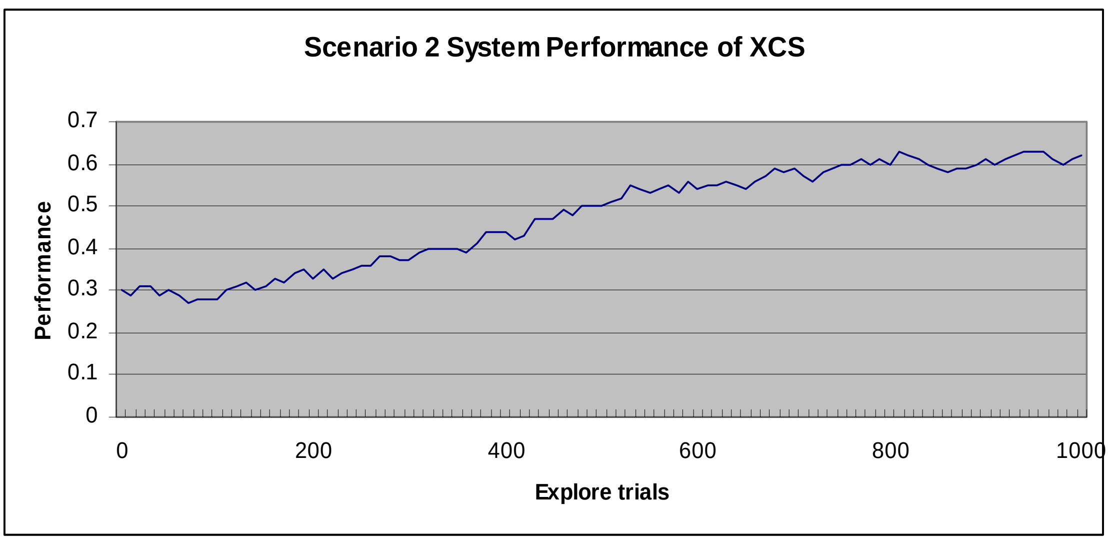


Figure 6.21. System Performance of XCS in Scenario 2

**6.6.2. Validation.** The degree of accuracy of the simulation models representing real system depends on model-related factors. These factors range from micro and macro parameters to initial conditions at the micro and macro level. The problem of validating models becomes more difficult when the simulation models contain non-linear relationships and randomness in individual behavior and interaction networks. Micro and macro variables that exhibit stochastic processes and feedback mechanisms between micro and macro levels add to the challenges associated with validating models that have these characteristics.

Xiang et al. (2005) summarize validation techniques used for simulation-based models. Internal validity compares the results of several replications of a stochastic simulation model using different random seeds. If model shows large variability under different random seeds, the model is questionable in terms of validity. Historical data validation is used when historical data is available. Part of the data is used to build the model, while the rest of the data is used to determine if the model behaves as the system does. Sensitivity analysis is another validation technique where parameters of the system are changed to determine the effect on the model and its output. Sensitive parameters are calibrated before using the model. Predictive validation is used for simulation models developed for prediction purposes. The model's prediction is compared with actual system behavior. Another technique that is more suitable for agent-based simulations is docking validation, which compares results of the simulation model to results of other models. In most cases, simulation models are compared with formal mathematical models developed for the same application. Agreement between models infers some degree of validity for the models. Docking methods do not completely validate the simulation model because formal models are also abstractions of the real system and are subject to empirical validation. Conducting statistical tests can increase the validity of the models. These tests can show whether simulation model behavior has an acceptable range of accuracy.

Fagiollo et al. (2006), outline three validation techniques specific for agent-based models. Indirect calibration method is a four-step approach to empirical validation. In the first step, modeler identifies a set of stylized facts that the model will reproduce. In the second step, the model is built based on empirical and experimental evidence. In the third step, the simulation is run to observe if the initial conditions yield the selected stylized facts. In the fourth step, empirical evidence on stylized facts is used to restrict the parameter space. This type of validation requires Monte Carlo techniques to sample the parameter space combination that will generate the stylized facts.

Midgley et al. (2006) utilize destructive testing for testing non-linear systems. This test combines ideas of extreme bounds, sensitivity analysis and robustness. They point out that an agent-based model can be validated through a higher level genetic algorithm to fit the model to empirical data. An objective function would be specified to reward closeness of fit to the empirical data.

The disadvantage of these empirical calibration approaches is that they reduce the space of possible states that are explored. This is especially true for agent-based models since these approaches limit the emergence properties of the models since the macro-system behavior is forced to replicate empirical results. As a result, it supports the continuation of current theories and models for which empirical data are available. Another criticism about these calibrating techniques is that a model with sufficient parameters can be adjusted to fit the data. For large multi-parameter models, there is no guarantee that the model is doing anything than curve fitting (Carley, 1994). However, for models where process is represented by rules, and not by parameterized equations this risk of curve fitting is less (Carley, 1994).

LeBaron (2006) suggests three steps for artificial stock market validation. First, the stock market should attempt to replicate difficult empirical features. This is often called analysis of stylized facts. Artificial time series properties are compared with real time series properties. If artificial stock market can generate properties observed in real markets, the model is validated to some degree. The second step is to put the parameters of the models under evolutionary control. This step often utilizes genetic algorithms to search parameter space to find better combinations of values. The third step is to use the results from laboratory experiments with human subjects to validate features of the model. Since agent-based models are an extension of experimental studies, the use of results from laboratory experiments can strengthen the model validation, but experimental studies are criticized in terms of not representing the real systems. Therefore, utilizing these results will partially validate the system.

Based on this review, the analysis of stylized facts is conducted for the artificial stock market simulations to partially validate the model. The statistical properties of the price series generated from the three scenarios are provided in Table 6.11. The statistical analyses show that in all the three scenarios, the price is not normally distributed because the kurtosis and skewness values are non-zero values. Scenario 1 and Scenario 2 have positive kurtosis values (leptokurtosis) which confirms that these markets also exhibit the fat tail phenomenon observed in real markets. The statistical values for the REE model are also provided for comparison in Table 6.11.

Table 6.11. Statistical Properties of the Experiments

| | **Kurtosis** | **Skewness** | **Mean** | **Standard Deviation** |
|---|---|---|---|---|
| Scenario 1 | 4.33 | -1.48 | 82.85 | 1.85 |
| Scenario 2 | 1.02 | -0.83 | 83.09 | 4.25 |
| Scenario 3 | -1.65 | 0.29 | 89.14 | 9.38 |
| REE | 0.19 | -0.22 | 72.93 | 5.06 |

Apart from the basic statistical characteristics of the artificial price series, the Jarque-Bera (JB) normality test is a test that measures the departure from normality based on the sample kurtosis and skewness. It is used to test the null hypothesis that data are from a normal distribution. The higher the JB statistics, the deviation from normality is higher. The Dickey-Fuller test shows whether a unit root is present in an autoregressive model. Time series are autoregressive models. If the unit root is present, the time series is said to have a stochastic trend. Table 6.12 provides the Jarque-Bera normality test for the three scenarios and Table 6.13 provides the Dickey-Fuller test results for the three scenarios. Appendix A provides the Jarque-Bera test formulas. Appendix A also provides the Dickey-Fuller table for analysis, as well as the coefficient and t-test results for the three scenarios. The Dickey-Fuller test reveals that all three scenarios tend to have a stochastic trend and are not linearly related.

Table 6.12. Jarque-Bera Normality Test Results

| | |
|---|---|
| Scenario 1 | 386.99 |
| Scenario 2 | 258.69 |
| Scenario 3 | 850.90 |

Table 6.13. Dickey-Fuller Test Results

| | |
|---|---|
| Scenario 1 | There is a unit root  with 98% confidence |
| Scenario 2 | There is a unit root with 98% confidence |
| Scenario 3 | There is a unit root with 98% confidence |

Another partial validation, model-to-model comparison (docking), is also conducted for this study. The three scenarios are compared with the Rational Expectation Equilibrium Model (REEM).

Figure 6.22 provides the output from the simulation for Scenario 1 where there is no bias mechanism. The graph also shows the REEM price for the same market conditions. The dark thick line represents the model price formation, whereas the thin line represents the REEM price formation for the same market structure. In Scenario 1, the model price and REEM price follow each other closely. The price dynamics are also similar where the model price increases as the REEM price increases. However, there are periods when the model price deviates from the REEM price. This is due to the traders using technical trading rules for making decisions. Even though traders have fundamental trading rules, the technical trading rules drive the market towards inefficient periods. There is also a difference between the variability of the model price and the REEM price. This is due to the pricing mechanism of the model. The alpha parameter analyzed in the sensitivity analysis section affects the variability of the model. Since REEM price is independent of alpha parameter, the REEM price is more variable, fluctuating between highs and lows.

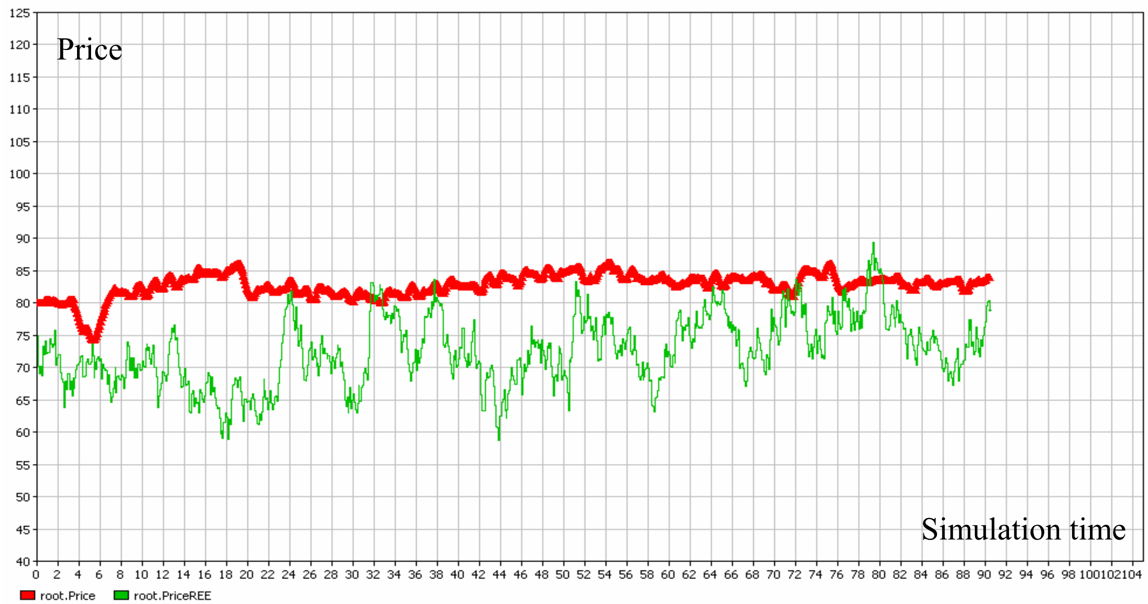


Figure 6.22. Comparison of Scenario 1 Price to the REEM Price

Figure 6.23 provides the output from the simulation for Scenario 2, including the REEM price. The dark thick line represents the model price formation, whereas the thin line represents the REEM price formation for the same market structure. In Scenario 2, at the beginning of the simulation, the difference between the model price and the REEM price is small. Also, the model price dynamics are similar to REEM price dynamics. As the REEM price increases, the model price also increases and vise versa. Towards the middle of the simulation, the model price starts to deviate from the REEM price. Also, the model price dynamics tends to move in an upward direction, whereas the REEM price dynamics swing between more highs and lows. Towards the end of the simulation, the model price gets close to the REEM price. In Scenario 2, there is a pattern in which there are periods where the difference between the REEM price and model price is not high, but there are also periods in which the difference between the REEM and model price fluctuates wildly. The market is not inefficient all the time, but goes through periods where it deviates from the efficient market model. In the model, the learning mechanism pulls the model price towards the REEM price, whereas the bias mechanism shifts the model price away from the REEM price.

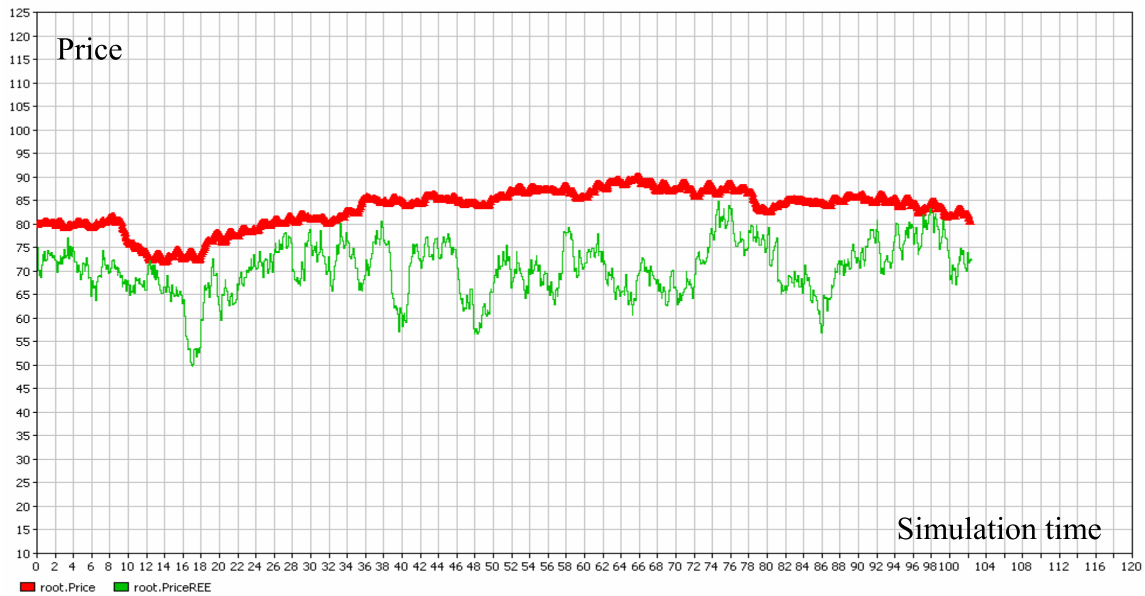


Figure 6.23. Comparison of Scenario 2 Price to the REEM Price

Figure 6.24 provides the output from the simulation for Scenario 3, including the REEM price. The dark thick line represents the model price formation, whereas the thin line represents the REEM price formation for the same market structure. In Scenario 3, the model price at some point in the simulation gets close to REEM price, but after some time the price drastically deviates from the REEM price. Also, the model price dynamics moves towards an upward trend, whereas the REEM price dynamics fluctuates. Since the bias mechanism dominates the trader behavior, the drastic deviation from the REEM price is an expected outcome. The learning mechanism is not successful enough to pull the market back to the REEM price dynamics.

## 6.7. SUMMARY

The trader architecture combines the Markov process based bias model and the XCS learning classifier based learning model into one hybrid model. The hybrid architecture is embedded into an agent-based financial market model to analyze the effect of the trader architecture on market dynamics. Three different scenarios are generated for the analysis. The analyses reveal that when a learning mechanism dominates the trader

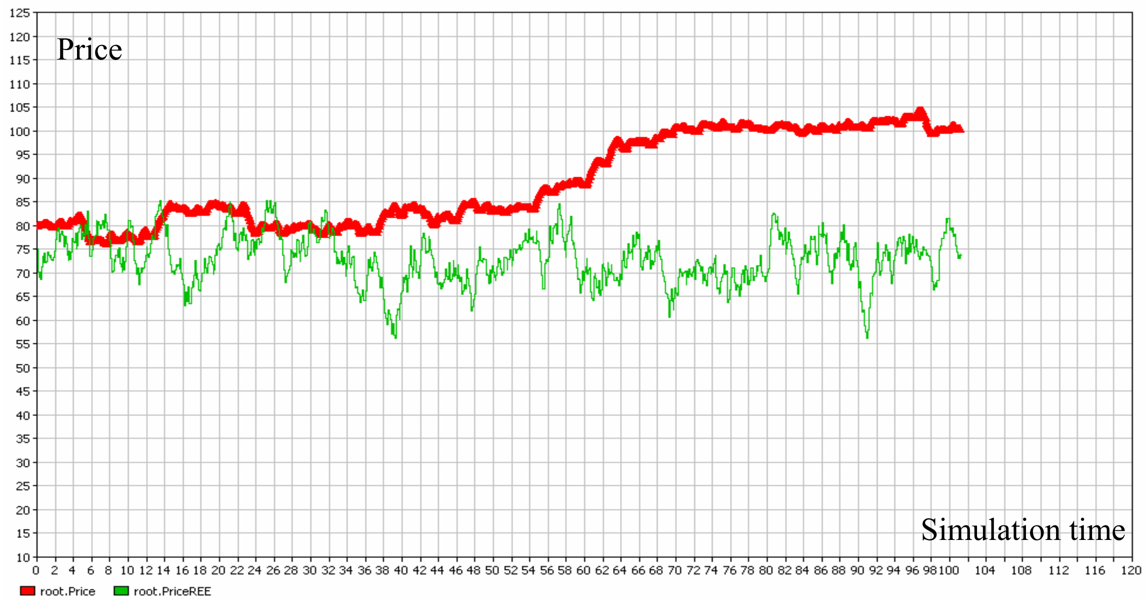


Figure 6.24. Comparison of Scenario 3 Price to the REEM Price

behavior, the model price and price dynamics closely follow the REEM price dynamics. For this case, the only factor for deviation from the REEM price is the use of technical trading rules. When the bias mechanism dominates the trader behavior, the model price and dynamics drastically deviate from the REE price dynamics. When both mechanisms function equally, the market shows a cyclic pattern where the market gets close to an efficient market and then shifts to an inefficient market. The adaptive model captures some of the real market properties. The study shows that there are two factors for deviation from the fundamental price. The use of technical trading rules deviates from the market price to a degree, but the drastic deviations from the fundamental price is due to the bias mechanism of traders. The bias mechanism under aggregate behavior has a significant affect on the market dynamics.

The study extends Barberis's behavioral model by incorporating it into an adaptive model. It also extends the artificial financial market models by creating a model that is based on cognitive characteristics of traders. Previous artificial financial market studies focused on distinguishing traders as fundamental and/or technical traders. The dominance of one or the other type of trader explained the cyclic patterns of efficient and inefficient markets. This model provides another perspective that traders can utilize both types of trading techniques, but the main deviation from efficient markets comes from their framing biases. This explains the reason why market dynamics depend on the information processing of traders.

This study illustrates that the Artificial Life framework can capture both the structural architecture of SoS as well as the social and mental structures of the system. This type of framework is especially beneficiary during what-if analysis of systems and can minimize cascading failures of systems by capturing different emergent behaviors of system architectures.

# 7. CONCLUSIONS AND FUTURE WORK

The need for better upper level descriptive and analysis frameworks is a challenging area in SoS studies. The motivation behind this study was to develop an analysis framework that integrates physical, information, cognitive and social components of SoS. The study presents Artificial Life based framework for modeling and analysis of emergent behavior of SoS architectures. The framework comprises cognitive architectures embedded in multi-agent models. Various computational intelligence tools can be utilized to design mechanism modules, which can be incorporated into the cognitive architecture. This type of framework provides a flexible and modular way of modeling sub-systems of System of Systems and captures the adaptive and emergent behavior of the system architecture. Specifically, a combination of deliberative and reactive reasoning provides a flexible architecture for modeling sub-systems of SoS. The meta-management level of the cognitive architecture plays an important role in designing evolutionary architectures. Additions and re-configuration of different modules, such as learning, long-term associative memory, short-term memory, and imitation can produce evolvable sub-systems.

Future studies of this framework should focus on the collaborative behavior between agents in the system. Interfaces are the leverages of complex systems. Therefore, the framework should provide means to capture different interfaces between sub-systems. The framework should also provide better means of modeling the Global Information Grid of the SoS meta-architecture. There are various models such as network theory, Internet network model, social network models that can be incorporated into the environment model. However, a more systematic approach should be developed to capture this component of the meta-architecture.

Many systems in nature deal with dynamically changing environment by forming swarm architectures. Swarm intelligence provides robust and scaleable solutions for interactions between agents. More focus should be given to the use of swarm intelligence for building SoS. Swarm intelligence can provide a solution to the scalability of cognitive agent architectures to larger system representations. Evolutionary modeling and learning are essential components of SoS. Current supervised learning techniques cannot handle rapidly evolving SoS. Reinforcement learning candidates, combined with other learning mechanisms, such as a supervised learning assisted reinforcement learning architecture or classifier systems, are more suitable for modeling system evolution in SoS. More emphasis should be given to development of hybrid learning mechanisms suitable for SoS operating environments.

Performance, risk, schedule and cost are some of the system attributes for comparing and selecting architectures. Ability to learn and evolve new architectures from the previously generated architectures, based on system attributes' values, needs to be incorporated in modeling and simulation process. The Artificial Life framework can become more valuable for system architects if performance attributes can be incorporated into the framework.

Both structural and object-oriented analysis is required for comprehension of SoS. Simulation tools that combine various modeling paradigms (discrete, agent-based, system dynamics) should be used in analysis of SoS to capture different behavioral views. The framework can be combined with other approaches to capture a better understanding of SoS.

To demonstrate the AL framework as an executable model, financial markets are selected as an analysis domain. The proposed financial market model is an agent-based model that exhibits adaptive behavior and combines adaptive learning with investor sentiment models. The proposed model was successful in integrating the trader mental processes, the physical market structure, and the aggregate trader behavior into one analysis model.

Previous studies of artificial stock market studies designed trader behavior in terms of simple buy/sell decisions. This study contributes to the artificial stock market studies by analyzing the effect of covering mechanism on market dynamics. In fact, under covering and learning mechanism, the stock market exhibited relatively less volatile dynamics and small deviation from rational expectations equilibrium price. Therefore, the covering mechanism turns out to be an important component in market system analysis. Previous artificial stock market studies utilized classical learning classifier systems as the adaptation mechanism. This study utilized XCS as a learning mechanism and eliminated some of the drawbacks of the original classifier system.

Learning classifier systems provided a combination of reinforcement learning and genetic algorithms. These types of hybrid learning mechanisms are necessary for modeling adaptation in continuously evolving environments. Biased strategies have not been incorporated into the classifier systems before. Most studies focused on fundamental and technical trading strategies and do not analyze the effect of biased strategies. This study contributed to the artificial stock market studies by analyzing if biased strategies survived against technical trading strategies. The analysis results revealed that biased strategies survived against other trading strategies and resulted in a market crash. Based on this result, hybrid trader architecture is proposed for the market analysis. The proposed architecture identified two mechanisms that play an important role in market dynamics: the bias mechanism and learning mechanism. The model captures a relationship between use of technical and fundamental trading rules under a learning mechanism and investor biases. Traders utilize fundamental and technical trading strategies to make decisions, but at some intervals are biased due to their imperfect heuristic rules. The bias model utilizes some of the key issues in behavioral finance. It expands behavioral finance studies by investigating the investor behavioral models under different market mechanisms and evolutionary conditions. The model shares the same view as behavioral finance and is designed to capture the potential dynamics leading to inefficient markets.

The proposed artificial stock market model is benchmarked against the rational expectations equilibrium model. The comparison provided insights about the deviation from equilibrium price. Previous artificial financial market studies focused on distinguishing traders as fundamentalists and technical traders. The dominance of one or the other type of trader explained the cyclic patterns of efficient and inefficient markets. This model provides another perspective that traders can utilize both types of trading techniques, but the main deviation from efficient markets comes from their framing biases.

In future studies, the affect of the proposed trader architecture should also be tested under different pricing mechanisms. In the artificial stock market models developed in this study, the price was increased or decreased by a constant amount based on the relationship between excess supply and demand. In future studies, the price formation can be determined by the intersection of the supply and demand curve or by the use of an automated auctioneer (market-maker). This can be more representative of real markets. In such a case, the artificial stock market can also be benchmarked against a real stock index.

The bias model incorporates two major heuristic driven biases to model under reaction and over reaction behavior of traders: representativeness and conservative behavior. The model does not incorporate saliency, overconfidence, and anchoring, but all these biases also lead to either over reaction or under reaction in markets. Future studies can focus on incorporating other bias models into the artificial financial market model to capture and analyze real market dynamics.

The proposed artificial stock market model is not designed for trader preference formation or the framing issues. The constant absolute risk aversion (CARA) assumption is used to simplify the model. This assumption is not realistic of real markets because risk level varies among traders and wealthier traders have more impact on the market. The variation in risk preferences should be included in some way. Loss aversion is the main factor that drives investors' tolerance for risk. Loss aversion is subject to framing. The model can be designed to add this property by incorporating Prospect Theory, which builds a relationship between levels of risk traders are willing to take based on loss aversion. The agent-based approach can expand the theory by analyzing the effect of various loss aversion values for traders. This study can give more insight to the risk premium puzzle.

The current artificial stock market model is not designed for trading interactions among traders. Agent-based frameworks provide the flexibility of modeling trader interactions. Therefore, if information transmission among traders is incorporated into the model, the model can also provide insight to the herding behavior observed in markets. This herding mechanism can provide insights to macro level volatility clustering behavior of markets. It can also give insight about why traders select similar trading strategies for certain market conditions.

Stock price is based on the market's expectation regarding the future. Therefore, in order to predict stock price, traders must form expectations that are better than the market's expectations. When forming expectations traders use a set of information and models for processing information. Therefore, to beat the market a trader must either have superior information, or process information that is better than others, or know behavioral biases leading to deviation from fundamental value. Fuller (1998) describes fundamentalists as analysts trying to capture superior information. He describes analysts that try to build better models to process information as quantitative analysts and describes analysts that try to exploit behavioral biases as behavioral analysts. Agent-based models can be used as tools to help behavioral analysts. If these studies are well calibrated, a hybrid model of agent-based and quantitative analysis can be utilized to better analyze the markets.

This work has developed a methodological approach to modeling and architecting system of systems and complex adaptive systems, specifically financial markets. The study comprised systems architecting, computational intelligence, modeling and simulation, financial market analysis and behavioral finance studies. The interdisciplinary characteristic of the study opens many unexplored aspects, which are briefly outlined in this section. Understanding and designing systems that can self organize and adapt without any outside control is the solution to successful System of Systems. How this can be achieved is the challenge that today's system engineers must face and solve.

# APPENDIX

## Jarque-Bera Test

The Jarque-Bera (JB) statistic test is defined as:

$$JB = \frac{n}{6}\left(S^2 + \frac{(K-3)^2}{4}\right)$$

where $n$ is the number of observations, $S$ is skewness and $K$ is kurtosis.

Skewness is defined as:

$$S = \frac{\frac{1}{n}\sum_{i=1}^{n}(x-\bar{x})^3}{\left(\frac{1}{n}\sum_{i=1}^{n}(x-\bar{x})^2\right)^{3/2}}$$

Kurtosis is defined as:

$$K = \frac{\frac{1}{n}\sum_{i=1}^{n}(x-\bar{x})^4}{\left(\frac{1}{n}\sum_{i=1}^{n}(x-\bar{x})^2\right)^{2}}$$

where $\bar{x}$ is the sample mean and n is the sample size.

## Dickey-Fuller Test

A simple auto-regressive model is:

$$y_t = \rho y_{t-1} + u_t$$

where $y_t$ is the variable of interest, t is the time index, $\rho$ is a coefficient and $u_t$ is the error term. The regression model can be re-written as:

$$\Delta y_t = (\rho - 1) y_{t-1} + u_t = \delta y_{t-1} + u_t$$

where $\Delta$ is the difference operator. Testing for a unit root can be done using the re-written form and this is equivalent to testing for δ=0. Since the test is done over the residual term rather than raw data, it is not possible to use standard t-distribution. Therefore, a specific distribution known as the Dickey-Fuller Table is used for critical values. The following table provides the Dickey-Fuller Table.

Dickey-Fuller Table

| | Sample Size | | | |
|---|---|---|---|---|
| | 25 | 50 | 100 | ∞ |
| F ratio (5%) | 7.24 | 6.73 | 6.49 | 6.25 |
| AR model with constant | | | | |
| 2% | -3.75 | -3.58 | -3.51 | -3.43 |
| 5% | -3.33 | -3.22 | -3.17 | -3.12 |
| 10% | -2.63 | -2.60 | -2.58 | -2.57 |
| AR model with constant and time trend | | | | |
| 2% | -4.38 | -4.15 | -4.04 | -3.96 |
| 5% | -3.95 | -3.80 | -3.69 | -3.66 |
| 10% | -3.24 | -3.18 | -3.15 | -3.13 |

The estimated slope coefficient from the regression and the t-statistics for all the there scenarios in Section 5 experiments are provided in the following table.

Regression statistics for experimental scenarios

| | **Slope coefficient (δ)** | **t-statistics** |
|---|---|---|
| Scenario 1 | -0.0103 | 0.5477 |
| Scenario 2 | 78.2031 | 374.8001 |
| Scenario 3 | 74.2966 | 301.7559 |

# BIBLIOGRAPHY

Adami C., (1998), "Introduction to Artificial Life," Springer-Verlag, New York.

Arthur W. B., Holland J., LeBaron B., Palmer R., and Tayler P., (1997), "Asset Pricing under Endogenous Expectations in an Artificial Stock Market," in *The Economy as an Evolving Complex System II*, Addison-Wesley, MA, pp. 15-44.

Axelrod R., (1984), "Evolution of Cooperation," Basic Books, New York.

Axelrod R., (1997), "Advancing the Art of Simulation in the Social Sciences: Simulating Social Phenomena," Springer, Berlin, pp. 21-40.

Axelrod, R., (1997), "The Complexity of Cooperation: Agent-based Models of Conflict and Cooperation," The Princeton University Press, Princeton, New Jersey.

Axtell, R., Axelrod, R., Epstein J., and Cohen, (1996), "Aligning Simulation Models: A Case Study and Results," *Computational and Mathematical Organization Theory*, Vol. 1 (2), pp. 123-141.

Axtell R., (2003), "The New Co-evolution of Information and Social Science: From Software Agents to Artificial Societies and Back or How More Computing Became Different Computing," *Working Paper, Brookings Institution*, http://www.econ.iastate.edu/tesfatsi/compsoc.axtell.pdf , October 2006.

Barberis N. and Thaler R., (2003), "A Survey of Behavioral Finance," in *Handbook of the Economics Finance*, pp. 1052-1090.

Barberis N., Shleifer A., Vishny R., (1998), "A Model of Investor Sentiment," *Journal of Financial Economics*, Vol. 49, pp. 307-343.

Bar-Yam Y., Allison M., Batdorf R., Chen H., Generazio H., Singh H. and Tucker S., (2004), "The Characteristics and Emerging Behaviors of System of Systems," *NECSI: Complex Physical, Biological and Social Systems Project*, http://necsi.org/education/oneweek/winter05/NECSISoS.pdf, May 2007.

Bar-Yam Y., (2003), "When Systems Engineering Fails-Toward Complex Systems Engineering," *2003 IEEE International Conference on Systems, Man & Cybernetics*, October 5–8, Washington, D.C., USA, pp. 2021-2028.

Bellamine N., Saoud B., (2002), "Modeling and Simulating Extreme Collaboration: An Agent-Based Approach," Invited Paper, COSI Summer School, July 2002, Crete, Greece. http://www.irit.fr/COSI/summerschool/bsecoll.pdf, March 2007.

Bernard V., (1992), “Stock Price Reactions to Earnings Announcements,” in *Advances in Behavioral Finance*, Thaler R. (ed.), Russell Sage Foundation, pp. 303-340.

Bruun C., (2001), “Chapter 1: Prospect for an Economics Framework for Swarm,” in *Agent-Based Methods in Economics and Finance: Simulations in Swarm,* Luna F. and Perrone A. (eds.), Kluwer Academic Publishers, Boston, pp. 187-227.

Bonabeau E., (2002), “Agent based Modeling: Methods and Techniques for Simulating Human Systems,” *Proceedings of the National Academy of Sciences*, Vol. 99, pp. 7280-7287.

Bonabeau E., and Meyer C., (2001), “Swarm Intelligence: A Whole New Way to Think about Business,” *Harvard Business Review*, May 2001, pp. 107-114.

Bonabeau E., Dorigo M., Theraulaz G., (1999), “Swarm Intelligence: From Natural to Artificial Systems,” Oxford University Press, New York.

Buda G., Choi D., Graveman R. and Kubic C., (2001), “Security Standards for the Global Information Grid,” *IEEE Military Communications Conference*, Vol. 1, pp. 617-621.

Bull L., Kovacs T., (eds.), (2005), “Foundations of Learning Classifier Systems,” Springer.

Burmeister B., (1996), “Models and Methodology for Agent-Oriented Analysis and Design,” in *Working Notes of the KI'96 Workshop on Agent-Oriented Programming and Distributed Systems*, K. Fischer (ed.), pp. 7 – 17, Document D-96-06. DFKI: Saarbrücken, Germany.

Bunn D., Oliveria F., (2001), "An Application of Agent-based Simulation to the New Electricity Trading Arrangements of England and Wales," *IEEE Transactions on Evolutionary Computation*, Vol. 5, Issue 5, pp. 493-503.

Butz M. V., and Wilson S. W., (2002), "An Algorithmic Description of XCS," *Journal of Soft Computing*, Vol. 6, pp. 144-153.

Calvano C.N., and John P., (2004), “Systems Engineering in an Age of Complexity,” *Systems Engineering Journal*, Vol. 7, Issue. 1, pp. 25-34.

Campbell J., Anderson D., Lawton C., Shirah D., and Longsine D., (2005), “System of Systems Modeling and Simulation,” *Proceedings of Conference on Systems Engineering Research 2005*, March 23-25, NJ, USA.

Carley K., (1996), “Validating Computational Models”, Working Paper: Social and Decision Sciences, Carnegie Mellon University, Pittsburgh, PA., http://www.econ.iastate.edu/tesfatsi/EmpValid.Carley.pdf, November 2006.

Carlock P.G., and Fenton R. E., (2001), "System-of-Systems (SoS) Enterprise Systems for Information-Intensive Organizations," *Systems Engineering*, Vol. 4, Issue 4, pp. 242–261.

Cazangi R., Zuben F., and Figueiredo M., (2003), "A Classifier System in Real Applications for Robot Navigation," *The 2003 Congress on Evolutionary Computation*, Vol. 1, Dec. 8-12, pp. 574-580.

Chan N., LeBaron B., Lo A. and Poggio T., (1999), "Agent-based simulation: Agent-based Models of Financial Markets: A Comparison with Experimental Markets," MIT Artificial Markets Project, Paper No. 124, http://citeseer.ist.psu.edu/chan99agentbased.html, March 2007.

Chen S., Yeh C., (2002), "On the emergent properties of artificial stock markets: the efficient market hypothesis and the rational expectations hypothesis," *Journal of Economic Behavior & Organization*, Vol. 49, pp. 217-239.

Chen X., (2003), "Co-evolutionary Multi-Agent-Based Modeling of Artificial Stock Market by Using the GP Approach," *IEEE International Conference on Computational Intelligence for Financial Engineering Proceedings*, pp. 159 -165.

Cherinka, R., Miller, R., and Smith, C., (2005), "Beyond Web Services: Towards On-Demand Complex Adaptive Environments," *IEEE International Conference on Web Services: (ICWS 2005): Proceedings*, July 11-15, Orlando, Florida, pp. 1-8.

Correa Y., and Keating C., (2003), "An Approach to Model Formulation for Systems of Systems," *IEEE Conference on Systems, Man and Cybernetics*, Vol. 4, pp. 3553-3558.

Crutchfield J. P., Mitchell M., and Das R., (2003), "The Evolutionary Design of Collective Computation in Cellular Automata," in Evolutionary Dynamics: Exploring the Interplay of Selection, Neutrality, Accident and Function, pp. 361-411. New York: Oxford University Press.

Dagli C., Kilicay N., (2007), "Understanding Behavior of System of Systems Through Computational Intelligence Techniques," CD Proceedings of 1st Annual IEEE Systems Conference, April 9-12, Honolulu, Hawaii.

De Long B., Scleifer A., Summers H., and Waldmann R., (1990), "Noise Trader Risk in Financial Markets," *The Journal of Political Economy*, Vol. 98, Issue 4, pp. 703-738.

Dermietzel J., Seese D., Stumpert T., (2006), "Market Clearing Prices in an Artificial Stock Market," *1st International Conference on Economic Sciences with Heterogeneous Interacting Agents*, July 2006, University of Bologna, Italy.

Dooley K., Corman S., (2000), "Agent-Based Genetic and Emergent Computational Models of Complex Systems," in *Encyclopedia of Life Support Systems*, Kiel L. D. (ed.), UNESCO/EOLSS Publishers, Oxford, U.K., http://www.eolss.net, March 2007.

Dumeur R., (1991), "Extended Classifiers for Simulation of Adaptive Behavior," *From Animals to Animats: Proceedings of the First International Conference on Simulation of Adaptive Behavior*, pp. 58-65.

Eberhart R., Kennedy J., (1995), "Particle Swarm Optimization," *Proceedings of the IEEE Conference on Neural Networks-Australia*, IV, pp. 1942-1948.

Edmonds, B., (1999), "What is Complexity? – The Philosophy of Complexity per se with Application to Some Examples in Evolution," in *The Evolution of Complexity*, Heylighen, F., and Aerts, D. (eds.), Kluwer, Dordercht.

Eiben A., and Smith J., (2003), "Introduction to Evolutionary Computing," Springer-Verlag Berlin Heidelberg New York.

Engelbrecht A., (2002), "Computational Intelligence: An Introduction," John Wiley & Sons Inc.

Epstein J., (2001), "Learning to be Thoughtless: Social Norms and Individual Computation," *Computational Economics*, Vol. 18, pp. 9-24.

Epstein J., Axtell R., (1996), "Growing Artificial Societies: Social Science from the Bottom-up," MIT Press, Cambridge, MA.

Fagiolo G., Windrum P., and Moneta A., (2006), "Empirical Validation of Agent-based Models: A Critical Survey," *LEM Working Paper 2006/14*, Laboratory of Economics and Management, Sant'Anna School of Advanced Studies, Pisa, Italy, http://www.lem.sssup.it/WPLem/files/2006-14.pdf , March 2007.

Fama E. F., (1970), "Efficient Capital Markets: A Review of Theory and Empirical Work," *Journal of Finance*, Vol. 25, pp. 383-417.

Fama E. F., (1991), "Efficient Capital Markets: II," *Journal of Finance*, Vol. 45, pp. 1575-1617.

Fama E. F., (1998), "Market Efficiency, Long-term Returns and Behavioral Finance," *Journal of Financial Economics*, Vol. 48, pp. 283-306.

Farnsworth G., Kelly J., Othling A., and Pryor R., (2004), "Successful Technical Trading Agents Using Genetic Programming," *SANDIA Report*, http://www.cs.sandia.gov/web9216/pubsagent/Markets.pdf , March 2007.

Ferber J., (1999), “Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence,” Addison-Wesley.

Flores-Mendez, R., (1999), “Towards a Standardization of Multi-Agent System Frameworks,” *ACM Crossroads*, Vol. 5, Issue 4, http://www.cpsc.ucalgary.ca./~robertof/index.html#publications, October 2006.

Focardi S., and Jonas C., (1997), “Modeling the Market: New Theories and Techniques,” Frank Fabozzi Associates, pp. 165-181.

Friedman D., and Sunder S., (1994), "Experimental Methods: A Primer for Economists," Cambridge University Press, pp. 132.134.

Fuller R., (1998), “Behavioral Finance and the Sources of Alpha,” *Journal of Pension Plan Investing*, Vol. 2, Issue 3, http://www.jpmorganfunds.com/pdfs/other/alpha.pdf, November 2006.

Gardner M., (1970), “Mathematical Games: The Fantastic Combinations of John Conway’s New Solitaire Game Life,” *Scientific American*, Vol. 223, pp. 120-123.

Gideon J., Dagli C., (2005), “Taxonomy of System-of-Systems,” *CD Proceedings of CSER Conference on Systems Engineering Research*, March 23-25, Hoboken, New Jersey.

Gilbert N., Troitzsch K. G., (1999), “Simulation for the Social Scientist,” Open University Press.

Gintis H., (2000), “Game Theory Evolving,” The Princeton University Press, Princeton, NJ.

Gode D. K., and Sunder S., (1993), "Allocative Efficiency of Markets with Zero-Intelligence Traders: Markets as a Partial Substitute for Individual Rationality," *The Journal of Political Economy*, Vol. 101, pp. 119-137.

Goldberg D. E., (1989), “Genetic Algorithms in Search, Optimization and Machine Learning,” Kluwer Academic Publishers, Boston, MA.

Griffin D., Tversky A., (1992), “The Weighing Evidence and the Determinants of Confidence,” *Cognitive Psychology*, Vol. 24, pp. 411-435.

Hall A.D., (1998), “Fractal Architecture of the Systems Engineering Method,” *IEEE Transactions on Systems, Man, and Cybernetics-Part C: Applications and Reviews*, Vol. 28, No. 4, pp. 565-572.

Heath C., and A. Tversky, (1991), “Preference and Belief: Ambiguity and Competence in Choice under Uncertainty,” *Journal of Risk and Uncertainty*, Vol. 4, pp. 5-28.

Hemsathapat K., Enke D., Dagli C., (2001), “Using a Neuro-Fuzzy-Genetic Data Mining Architecture to Determine a Marketing Strategy in a Charitable Organization’s Donor Database,” *CD Proceedings of 2001 IEEE International Engineering Management Conference.*

Holland J., (1975), “Adaptation in Natural and Artificial Systems,” University of Michigan Press, Ann Arbor.

Holland J., (1995), “Hidden Order: How Adaptation Builds Complexity,” Addison-Wesley, New York.

Hommes C. H., (2002), "Modeling the Stylized Facts in Finance through Simple Nonlinear Adaptive Systems," *PNAS*, Vol. 99, pp. 7221-7228.

Iglesias C., Garijo M., Gonzales J., (1998), “A Survey on Agent-Oriented Methodologies,” *Proceedings of the 5th International Workshop on Intelligent Agents: Agent theories, Architectures and Languages*, in LNAI, Vol. 1555, Springer.

Janssen M., (ed.), (2002), “Complexity and Ecosystem Management: The Theory and Practice of Multi-Agent Systems,” Elgar, London.

Jiao W., Zhou M., and Wang Q., (2003), “Formal Framework for Adaptive Multi-Agent Systems,” *Proceedings of the IEEE/WIC International Conference on Intelligent Agent Technology*, pp. 442-445.

Jennings N., Lesperance Y., (2000), “Lecture Notes in Artificial Intelligence, Intelligent Agents VI, Agent Theories, Architectures and Languages,” Springer.

Johnson C., (2003), “Future Combat Systems,” *Swarming: Network Enabled C4ISR Conference Proceedings*, Section D., pp. 11-27.

Jones R. M., (2004), “An Introduction to Cognitive Architectures for Modeling and Simulation,” *Interservice/Industry Training, Simulation and Education Conference*, http://www.soartech.com/pubs/tutorial-iitsec-2004.PDF, May 2007.

Joshi S., and Bedau M., (1998), “An Explanation of Generic Behavior in an Evolving Financial Market,” in *Complex Systems 98: Complexity Between The Ecos, From Ecology to Economics*, R. Standish, B. Henry, S. Watt, R. Marks, R. Stocker, D. Green, S. Keen and T. Bonssomaier (eds.), Complexity Online Network, Sydney, pp. 327-335, http://citeseer.ist.psu.edu/372008.html, March 2007.

Joslyn C., and Roca L., (2000), “Towards Semiotic Agent-based Models of Socio-Technical Organizations,” *Proc. AI, Simulation and Planning in High Autonomy Systems Conference*, Tucson, Arizona, pp. 70-79.

Kahneman D., and Tversky A., (1979), "Prospect Theory: An Analysis of Decision under Risk," *Econometrica,* Vol. 47, pp. 263-291.

Kaipei L., Yuhang H., Lanfang L., (2002), "Hybrid Multi-agent Based Modeling for MIS in Electric Power Enterprise, Power System Technology," *Proceedings of PowerCon 2002. International Conference on*, October 13-17, Vol. 3, pp. 1975-1979.

Kauffman, S., (1995), "At Home in the Universe: The Search for the Laws of Self-Organization and Complexity," Oxford Press, New York.

Kilicay N., Dagli C., (2003a), "Emergent Phenomena and Simulating Human Systems," *Proceedings of the 24th Annual Conference, American Society for Engineering Management*, St.Louis, Missouri, pp. 233-237.

Kilicay N., Dagli C., (2003b), "Emergence and Artificial Life," *IEEE Engineering Management Conference*, Albany, New York, pp. 580-584.

Kilicay N., Enke D., Ramakrishnan S., and Dagli C., (2005), "A Study of Artificial Financial Market," *Intelligent Engineering Systems through Artificial Neural Networks*, Vol. 15, pp. 625-634.

Kilicay N., Dagli C., and Enke D., (2006a), "Multi-agent Architectures for Analysis of Complex Adaptive Systems," *Proceedings of the 7th International Conference, Adaptive Computing in Design and Manufacture*, Bristol, UK, pp. 185-190.

Kilicay N., Enke D., Ramakrishnan S., and Dagli C., (2006b), "Trader Behavior Under an Evolving Stock Market Environment," *Intelligent Engineering Systems through Artificial Neural Networks*, Vol. 16, pp. 773-778.

Kilicay N., Dagli C., (2007a), "Methodologies for Understanding Behavior of System of Systems," *CD Proceedings of Conference on System Engineering Research*, March 14-16, Hoboken, New Jersey.

Kilicay N., Enke D., Dagli C., (2007b), "Analysis of System Behavior through Cognitive Architectures," *Proceedings of International Conference on Artificial Intelligence, Extending Computational Cognitive Modeling to Multi-agent Interaction*, June 25-28, Las Vegas, Nevada.

Kubik A., (2003), "Toward a Formalization of Emergence," *Artificial Life Journal*, Vol. 9, pp. 41-65.

Koza J. R., (1992), "Genetic Programming: On the Programming of Computers by Means of Natural Selection," MIT Press.

Langley P., Laird J., Rogers S., (2006), "Cognitive Architectures: Research Issues and Challenges," Working Paper, http://cll.stanford.edu/~langley/papers/final.arch.pdf, May 2006.

Langton C., (ed), (1989), "Artificial Life," Addison-Wesley, MA.

Lansig J., (2002), "Artificial Societies and the Social Sciences," *Artificial Life Journal*, Vol. 8, pp. 279-292.

La Porta R., (1996), "Expectations and the Cross-section of Returns," *Journal of Finance*, Vol. 51, pp. 1715-1742.

LeBaron B., Arthur W., and Palmer R., (1999), "Time Series Properties of an Artificial Stock Market," *Journal of Economic Dynamics and Control*, Vol. 23, pp. 1487-1516.

LeBaron B., (2000), "Agent-based Computational Finance: Suggested Readings and Early Research," *Journal of Economic Dynamics and Control*, Vol. 24, pp. 679-702.

LeBaron B., (2001), "Evolution and Time Horizons in Agent-Based Stock Market," *Macroeconomic Dynamics*, Vol. 5, pp. 225-254.

LeBaron B., (2002a), "Short-memory Traders and Their Impact on Group Learning in Financial Markets," *Proceedings of National Academy of Sciences*, Vol. 99, Issue 3, pp. 7201-7206.

LeBaron B., (2002b), "Building the Santa Fe Artificial Stock Market," *Working paper, Brandeis University*, http://www.econ.iastate.edu/tesfatsi/blake.sfisum.pdf, March, 2007.

LeBaron B., (2006), "Agent-Based Computational Finance," in *The Handbook for Computational Economics, Vol. II*, Judd & Tesfatsion (eds.), Elsevier/North-Holland.

Lettau M., (1997), "Explaining the Facts with Adaptive Agents: The Case of Mutual Fund Flows," *Journal of Economic Dynamics and Control*, Vol. 21, pp. 1117-1147.

Levy D., (2000), "Applications and Limitations of Complexity Theory in Organizational Theory and Strategy," *Handbook of Strategic Management*, pp.67-87, Marcel Decker, New York.

Lux T., Marchesi M., (1999), "Scaling and Criticality in a Stochastic Multi-agent Model of a Financial Market," *Nature*, Vol. 397, Issue Feb. 11, pp. 493-497.

Madhavan., A., (2000), "Market Microstructure: A Survey," *Journal of Financial Markets*, pp. 205-258.

Macy M. W., Willer R., (2002), “From Factors to Actors: Computational Sociology and Agent-Based Modeling,” *Annual Review of Sociology*, Vol. 28, pp. 143-166.

Madni A. M., Sage A. P., and Madni C., (2005), “Infusion of Cognitive Engineering into Systems Engineering Processes and Practices,” *IEEE International Conference on Systems, Man and Cybernetics*, Vol. 1, October 10-12, pp.960-965.

Madwaraj R., Ramakrishnan S., Dagli C., and Miller A., (2006), “Modeling the Global Earth Observation System of Systems,” *Industrial Engineers Research Conference*, Orlando, Florida.

Maes P., (1994), “Modeling Adaptive Autonomous Agents,” *Artificial Life Journal*, Vol.1, Issue 1, pp. 135-162.

Maier M. W., (2005), “Research Challenges of System-of-Systems,” *IEEE International Conference on Systems Man and Cybernetics*, Vol. 4, October 10-12, pp. 3149-3154.

Marchesi M., Cincotti S., Focardi S. and Raberto M., (2000), “Development and Testing of an Artificial Stock Market,” *Proceedings of Modelli Dinamici in Economia e Finanza*, Urbino, http://www.smartquant.com/references/MultiAgent/agent3.pdf, March 2007.

Margolus N., Toffoli T., (1987), “Cellular Automata Machines: A New Environment for Modeling,” MIT Press.

Matthews D., Collier P., (2000), “Assessing the Value of a C4ISREW System of Systems Capability,” *Proceedings of the 5th International Command and Control Research and Technology Symposium*, Canberra, Australia.

McCulloch W. S., (1965), “Embodiments of Mind,” MIT Press, Cambridge, MA.

Midgley D., Marks R., and Kunchamwar D., (2007), “The Building and Assurance of Agent-Based Models: An Example and Challenge to the Field,” *Journal of Business Research*, forthcoming, http://www.agsm.edu.au/~bobm/papers/jbr-mmk.pdf , March 2007.

Mitchell M., Forrest S., (1998), “Genetic Algorithms and Artificial Life,” *Artificial Life Journal*, Vol. 1, Issue 3, pp. 267-289.

Mitchell, M., (1996), “An Introduction to Genetic Algorithms,” MIT Press, Cambridge, MA.

Mitleton-Kelly E., (2003), “Ten Principles of Complexity and Enabling Infrastructures,” Complex Systems and Evolutionary Perspectives on Organizations: The Application of Complexity Theory to Organizations, Elsevier, pp.1-31.

Mobley M. D., Dagli C., (2006), “Heuristics and Genetic Algorithms,” *CD Proceedings of INCOSE 2006*, July 19-23, Orlando, Florida.

Mondada F., Guignard A., Bonani M., Bar D., Lauria M., and Floreano D., (2003), “SWARM-BOT: From Concept to Implementation,” *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vol. 2, pp. 1626 – 1631.

Nicolaisen J., Petrov V., Tesfatsion L., (2001), “Market Power and Efficiency in a Computational Electricity Market with Discriminatory Double-auction Pricing,” *IEEE Transactions on Evolutionary Computation*, Vol. 5, pp. 504-523.

Nicolis G., Prigogine I., (1989), “Exploring Complexity,” W. H. Freeman and Company, New York.

Nwana S., (1996), “Software Agents: An Overview,” *Knowledge Engineering Review*, Vol. 11, Issue 3, pp. 205-244.

Olson E., (1997), “The Ontological Basis of Strong Artificial Life,” *Artificial Life Journal*, Vol. 3, pp. 29-39.

Padberg J., Weber H. and Sunbul A., (2002), “Petri Net Based Components for Evolvable Architectures,” *Journal of Integrated Design and Process Science*, Vol. 6, pp. 105-114.

Palmer R., Arthur B., Holland J. and LeBaron B., (1994), “Artificial Economic Life: A Simple Model of a Stock Market,” *Physica D*, Vol. 75, pp. 264-274.

Parunak H., Savit R., and Riolo R., (1998), "Agent-based Modeling vs. Equation-Based Modeling: A Case Study and Users' Guide," *Proceedings of Multi-agent Systems and Agent-based Simulations*, LNAI 1534, Springer, pp. 10-25.

Parunak H., (2000), “The Process-Interface-Topology Model: Overlooked Issues in Modeling Social Systems,” *Proceedings of Modeling Artificial Societies and Hybrid Systems*, http://www.erim.org/~vparunak/masho00.pdf, May 2007.

Pascale R.T., Millemann M., and Gioja L., (2000), “Surfing the Edge of Chaos: The Laws of Nature and the New Laws of Business,” Three Rivers Press, New York.

Pfister A., (2003), “Heterogeneous Trade Intervals in an Agent Based Financial Market,” *SFB Adaptive Information Systems and Modeling in Economics and Management Science*, http://www.wu-wien.ac.at/am/Download/wp99.pdf, March 2007.

Poggio T., Lo A., LeBaron B., Chan T., (2001), "Agent-based Models of Financial Markets: A Comparison with Experimental Markets," MIT Sloan Working paper No. 4195-01, http://people.brandeis.edu/~blebaron/wps/disagg.pdf , May 2007.

Rechtin E., and Maier M., (1997), "The Art of Systems Architecting," CRC Press, Boca Raton, Florida.

Ronald E., Sipper M., Capcarrere M., (1999), "Design, Observation, Surprise! A Test of Emergence," *Artificial Life Journal*, Vol. 5, pp. 225-239.

Ross S., (2005), "Neoclassical Finance," Princeton University Press.

Russell S., and Norvig P., (2003), "Artificial Intelligence: A Modern Approach," Prentice Hall.

Sage A. P., and Cuppan. C. D., (2001), "On the Systems Engineering and Management of Systems-of-Systems and Federations of Systems," *Information, Knowledge, Systems Management*, Vol. 2, Issue 4, pp. 325–45.

Sanchez S., Lucas T., (2002), "Exploring the World of Agent-Based Simulations: Simple Models, Complex Analyses," *Proceedings of the 2002 Winter Simulation Conference*, eds. J. L. Snowdon, J. Charnes, C.-H. Chen, and E. Yucesan, Institute of Electrical and Electronic Engineers: Piscataway, NJ, pp. 116-126.

Schlagel R., (1999), "Why not Artificial Consciousness or Thought?," *Minds and Machines*, Vol. 9, pp. 3-28.

Schoreels C., Logan B., and Garibaldi J., (2004), "A Preliminary Investigation into Multi-agent Trading Simulations Using a Genetic Algorithm," *Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology*, Beijing China, pp. 421-424.

Schulenburg S. and Ross P., (2000), "An Adaptive Agent Based Economic Model," *Learning Classifier Systems: From Foundations to Applications*, Springer-Verlag, Berlin, Vol. 1813, pp. 265-284.

Schleifer A., (2000), "Inefficient Markets: An Introduction to Behavioral Finance," Clarendon Lectures in Economics, Oxford Press, pp. 112-153.

Semet Y., Gelly S., Schoenauer M., and Sebag M., (2004), "Artificial Agents and Speculative Bubbles," *CF'04: International Conference on Computational Finance & its Applications (PKDD-2004)*, Pisa, Italy.

Shiller, R. J., (2003), "From Efficient Market Theory to Behavioral Finance," *Journal of Economic Perspectives,* Vol. 17, pp. 83-104.

Shefrin H., (2002), “Beyond Greed and Fear: Understanding Behavioral Finance and the Psychology of Investing,” Oxford University Press.

Simon, H., (1969), "The Sciences of the Artificial," MIT Press, Cambridge, MA.

Sloman A., Logan B., (2000), “Evolvable Architectures for Human-Like Minds,” in *Affective Minds*, G. Hatano, N. Okada and H. Tanabe (eds.), Elsevier, Amsterdam.

Sloman A., Scheutz M., (2002), “A Framework for Comparing Agent Architectures,” *Proceedings of UK Workshop on Computational Intelligence, Birmingham*, http://www.cs.bham.ac.uk/~ukci, October 2006.

Smith R., Dike D., Mehra R., Ravinchandran B., and El-Fallah A., (1999), “Classifier Systems in Combat: Two-sided Learning of Maneuvers for Advanced Fighter Aircraft,” in *Computer Methods in Applied Mechanics and Engineering*, Elsevier, Amsterdam.

Sunghwan S., Dagli C., (2001), “Advantages of Using Fuzzy Class Memberships in Self-Organizing Map and Support Vector Machines,” *CD Proceedings of International Joint Conference on Neural Networks*, July 14-19, Washington, DC, pp. 1886-1890.

Stanilka S. P., Miller A., Dagli C., (2005), “Object-Oriented Development for DoDAF System of Systems,” *CD Proceedings of INCOSE 2005*, July 10-14, Rochester, New York.

Steels L., (1994), “The Artificial Life: Roots of Artificial Intelligence,” *Artificial Life Journal*, Vol. 1, Issue 1, pp. 89-125.

Sycara K., (1998), “Multi-agent Systems,” *Artificial Intelligence Magazine*, Vol. 10, Issue 2, pp. 79-93.

Takashi H., and Terano, T., (2002), "Agent-based Approach to Investors' Behavior and Asset Price Fluctuation in Financial Markets," *Journal of Artificial Societies and Social Simulation*, Vol. 6, Issue 3, http://jasss.soc.surrey.ac.uk/6/3/3.html, March 2007.

Tan N. and Lin S., (2001), “Fuzzy Inductive Reasoning, Expectation Formation and the Behavior of Security Prices,” *Journal of Economic Dynamics and Control*, Vol. 25, pp. 321-362.

Tassier T., Everson M., Ostrowski D., (2002), “Agent-Based Models as a Complement to Economic Theory: A Durable Goods Example,” *IEEE Proceedings of the 2002 Congress on Evolutionary Computation*, pp. 729-734.

Taylor J. H., Sayda A., (2005), "An Intelligent Architecture for Integrated Control and Asset Management for Industrial Processes," *Proceedings of the IEEE International Symposium on Intelligent Control*, Limossol, Cyprus, June 27-29, pp. 1397-1404.

Terna P., (2001), "Chapter 8: Cognitive Agents Behaving in a Simple Stock Market Structure," in *Agent-Based Methods in Economics and Finance: Simulations in Swarm*, Luna F. and Perrone A. (eds.), Kluwer Academic Publishers, Boston, pp. 187-227.

Tesfatsion L., (2001a), "Economic agents and markets as emergent phenomena," *Proceedings of the National Academy of Sciences*, Vol. 99, Issue 3, pp. 7191-7192.

Tesfatsion L., (2001b), "Agent-based Computational Economics: Growing Economies from the Bottom Up," *Artificial Life*, Vol. 8, Issue 1, pp. 55-82.

Thalmann N., Thalmann D., (1994), "Artificial Life and Virtual Reality," John Wiley & Sons Ltd.

Toffoli T. and, Margolus N., (1987), "Cellular Automata Machines: A New Environment for Modeling," MIT Press.

Umheh N., Miller A., Dagli C., (2007), "TOGAF vs. DoDAF: Architecting Frameworks for Net-Centric Systems," *CD Proceedings of IERC Industrial Engineering Research Conference*, May 19-23, Nashville, Tennessee.

Vriend N., (2000), "An Illustration of the Essential Difference between Individual and Social Learning, and its Consequences for Computational Analysis," *Journal of Economic Dynamics & Control*, Vol. 24, pp. 1-94.

Waldrop M. M., (1992), "Complexity: The Emerging Science at the Edge of Order and Chaos," Touchstone, New York.

Williams B. J., Campbell K., and McKay M. D., (2005), "Sensitivity Analysis When Model Outputs are Functions," in *Sensitivity Analysis of Model Output*, Kenneth M H., and Francois M. H. (eds.) Los Alamos National Laboratory.

Wilson S. W., (1991), "The Animat Path to AI," in *From Animals to Animats*, Meyer J. A. and Wilson S. (eds), pp. 15–21.

Wilson S. W., (1995), "Classifier Fitness Based on Accuracy," *Evolutionary Computation*, Vol.3, Issue 3, pp. 149-175.

Wolfram S., (2002), "A New Kind of Science," Wolfram Media Inc.

Wooldridge M., Jennings N. R., Kinny D., (2000), “The Gaia Methodology for Agent-Oriented Analysis and Design,” *Autonomous Agents and Multi-Agent Systems*, Vol. 3, Issue 3, pp. 285 – 312.

Xiang X., Kennedy R., and Madey G., (2005), “Verification and Validation of Agent-based Scientific Simulation Models,” *Agent-directed Simulation Conference*, San Diego.

Yeh C., and Chen S., (2002), “Toward and Integration of Social Learning and Individual Learning in Agent-Based Computational Stock Markets: The Approach Based on Population Genetic Programming,” *Journal of Economic Behavior & Organization*, Vol. 49, Issue 2, pp. 217-239.

Youssefmir M., Huberman B., (1997), “Clustered Volatility in Multi-agent Dynamics,” *Journal of Economic Behavior & Organization*, Vol. 32, pp. 101-118.

# VITA

Nil Hande Kilicay-Ergin was born in Istanbul, Turkey on July 28th 1977. She completed her Bachelor of Science degree in Environmental Engineering at Istanbul Technical University, Istanbul, Turkey in June 1999. She received her Master of Science degree in Engineering Management from the University of Missouri at Rolla in December 2001. In August 2002, she entered the PhD program and earned her PhD degree in Systems Engineering from the University of Missouri at Rolla in June 2007. While she was pursuing her PhD degree, she worked as a research and teaching assistant at the departments of Engineering Management and Systems Engineering. She was also associated with Smart Engineering Systems Lab and Laboratory for Investment and Financial Engineering at the University of Missouri – Rolla. Her research interests are in modeling and analysis of complex adaptive systems, system of systems, systems architecting, artificial life, financial markets and behavioral finance.