AN AGENT-BASED NETWORK
THEORETICAL APPROACH TO A
COMPLEX ADAPTIVE COMPUTER
MANUFACTURING ENTERPRISE
ECOSYSTEM

A Thesis Presented

By

Selen Onel

To

Department of Mechanical and Industrial Engineering

In Partial Fulfillment for the Requirements
For the Degree of

Doctor of Philosophy

In the field of

Industrial Engineering

NORTHEASTERN UNIVERSITY
Boston, Massachusetts

May 2012
Abstract

In this dissertation we investigate the topological and dynamic characteristics of computer manufacturing enterprises needing to operate competitively in a rapidly changing business environment. Today business markets are highly non-linear, fast evolving and emerging. Crucial part of this dynamic structure is supply chains with innumerable interactions and inter-dependencies, altering global and local economies, and increasing customer expectations. In this sheer complexity and unpredictability enterprises look for different approaches to have more agile supply chains. They adopt different strategies to anticipate and quickly adapt to forthcoming changes. They are sensitive to environmental disturbances. In order to survive, they are in need of strong information flow and collaboration links with their suppliers, customers and even surprisingly with their current and potential competitors.

In previous studies, researchers had studied supply chains using equation-based approaches, traditional simulation based approaches, or a combination of both. However, these modeling approaches are limiting to capture the dynamics of today’s complex adaptive supply chain systems. Therefore, in this research, we conceive and analyze a realistic computer simulated computer manufacturing enterprise ecosystem (CMEE) that provides insights into the behavior of enterprises that ought to be highly dynamic, scalable, reconfigurable, agile and adaptive.

We study the evolution of topological structures that form the computer manufacturing enterprise ecosystem by using complex adaptive systems theory perspective using an agent based model approach and network theoretic methods. The agent based model is built based on the self-
organization and adaptation concepts borrowed from biological ecosystems and agility and alignment concepts from manufacturing systems. This study explores the underlying complexity and dynamism of the computer manufacturing market. Further it investigates the evolution of topological structures and dynamic behaviors that form the CMEE. It studies the influence and effect of decisions and actions of enterprises and changing trends in computer manufacturing environment.

The results of this dissertation showed that enterprises, which adjust goals and infrastructure quickly according to the changes in the customers, suppliers, and/or competitors survive longer in the CMEE compared to enterprises that resist change. Also, according to our findings enterprises, which adopt a cooperative behavior, benefit from increased collaboration and information flows with their neighbors and gain advantage over their greedy competitors. The findings also show that changes in environment condition and difficulty level of growing and branching out in the CMEE have statistically significant effects in market life expectancy of the CMEE.

Companies can use the outcomes of this project to build agile, robust, and adaptive enterprises that anticipate, transform, and thrive in harsh business environment conditions. The current project focuses primarily on desktop and laptop computer related companies as test beds to develop and demonstrate the approach; however the approaches and methodologies develop herein are broadly applicable to industries such as consumer electronics and automobiles sectors that operate in rapidly changing business environment.
Acknowledgements

I am glad to express my gratitude to my thesis advisors Dr. Sagar V. Kamarthi and Dr. Abe Zeid for the infinite patience, and incredible amount of time they put in directing this dissertation. Besides the constant review progress of my research work at every stage of developing new methodologies and algorithms presented in this thesis, Dr. Kamarthi and Dr. Zeid paid attention to the tiniest level of details in correcting the text. They have been a great source of my motivation. I will always be deeply indebted to them for teaching me how to be a good researcher and a good writer. A magnificent achievement such as completing this thesis would not have been possible without their encouragement and support in all aspects.

I am extremely delighted to express my gratitude to my dissertation committee members; Dr. Ginestra Bianconi and Dr. Mark A. Kon. I would also like to thank you Dr. Albert-László Barabási for accepting to be an honorary member in my dissertation committee despite his busy schedule.

I was funded with graduate assistantships from the Undergraduate School of Engineering and Graduate School of Engineering, the Department of Mechanical and Industrial Engineering. I am grateful to Dr. Richard Scranton, Dr. David Kaeli, Dr. Rachelle Reisberg, Dr. Yaman Yener, Dr. Emanuel Melachrinoudis, and Dr. Hameed Metghalchi for providing the assistantship and additional summer funding.
I would like to thank the faculty and staff of the College of Engineering for their general encouragement and support. My special thanks are also due to Noah Japhet, Joyce M. Crain and Jonathan Doughty. Thanks, also, to Le Baron Briggs, Lisa Koch, James Jones, Tom Papadopoulos and David Moore for their help in meeting my computing needs. I owe a great deal of thanks to the staff of Inter Library Loan (ILL) of Snell Library.

I have been happy in the company of my lab mates and colleagues Yusuf Ozbek, Amal Al-Husseini, Emre Tuncel, Jessica Chin, Tony Sultornsanee, Hatem Abuhimd, Srinivasan Radhakrishnan, Arjun Duvvuru, Nimesh Patel, Meet Shah, Ghulam Moeen Uddin, Ilke Boyaci and Abdullah Polat.

At last but not least, I am forever indebted to my beloved friends and family whose help, love, and encouragement is incomparable and unforgettable. My stay in America has been a pleasure, fun, adventure, personal growth and comfortable by the support, generosity and love of my friends, MetuCapoeiraFriends, my Sinha Capoeira family and my family. Reaching America to do research would not have been possible without the love, care, concern, and foresight of my parents Terlan Asli Onel and Yucel Onel and my sister Selis Onel. I owe them a large portion of all the good things that life has given me. I am grateful to them forever. As a final remark, I also would like to thank you everyone, whom I might forget to mention their names that has helped me and made a contribution to this big achievement in my life.

Selen Onel
March 27, 2012
Boston, MA 02115, USA
## Table of Contents

Abstract ................................................................................................................................................. 3

Acknowledgements ............................................................................................................................... 5

Table of Contents .................................................................................................................................. 7

Chapter 1 - Introduction .......................................................................................................................... 16

1.1. Overview .......................................................................................................................................... 18

1.2. Research Objectives ......................................................................................................................... 21

1.3. Motivation ......................................................................................................................................... 22

1.4. Problem Statement ............................................................................................................................ 23

1.5. Solution ............................................................................................................................................ 24

1.6. Contribution ..................................................................................................................................... 26

Chapter 2 - Literature Review .................................................................................................................... 28

2.1. Introduction to Computers and Computer Components ................................................................. 33

2.2. Nature of Computer Manufacturing Industry ................................................................................. 39

2.3. Organization of Computer Manufacturing Industry ....................................................................... 41

Chapter 3 - Complex Systems .................................................................................................................. 44

3.1. Why Complex Systems? .................................................................................................................. 46

3.2. Three Approaches to the Study of Complex Systems ..................................................................... 47

3.3. Types of Complex Systems .............................................................................................................. 47

3.3.1. Chaotic Systems ......................................................................................................................... 47

3.3.2. Complex Adaptive Systems ....................................................................................................... 48

3.3.3. Nonlinear System ....................................................................................................................... 48
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.4. Features of Complex Systems</td>
<td>48</td>
</tr>
<tr>
<td>3.5. Complex Adaptive Systems</td>
<td>50</td>
</tr>
<tr>
<td>3.6. Features of Complex Adaptive Systems</td>
<td>51</td>
</tr>
<tr>
<td>3.6.1. Agency</td>
<td>51</td>
</tr>
<tr>
<td>3.6.2. Self-organization</td>
<td>52</td>
</tr>
<tr>
<td>3.6.3. Emergence</td>
<td>52</td>
</tr>
<tr>
<td>3.6.4. Network Connectivity</td>
<td>52</td>
</tr>
<tr>
<td>3.6.5. Dimensionality and Feedback</td>
<td>53</td>
</tr>
<tr>
<td>3.6.6. Dynamism</td>
<td>54</td>
</tr>
<tr>
<td>3.6.7. Adaptivity</td>
<td>54</td>
</tr>
<tr>
<td>3.6.8. Rugged Landscape</td>
<td>55</td>
</tr>
<tr>
<td>3.6.9. Co-evolution</td>
<td>55</td>
</tr>
<tr>
<td>3.6.10. Quasi-equilibrium</td>
<td>56</td>
</tr>
<tr>
<td>3.6.11. Non-linearity</td>
<td>57</td>
</tr>
<tr>
<td>3.6.12. Non-random Future and Path Dependence</td>
<td>57</td>
</tr>
<tr>
<td>3.6.13. Difficulty to Determine Boundaries</td>
<td>58</td>
</tr>
<tr>
<td>Chapter 4 - Network Theory</td>
<td>59</td>
</tr>
<tr>
<td>4.1. Modeling of Complex Networks</td>
<td>60</td>
</tr>
<tr>
<td>4.1.1. Random Graphs</td>
<td>60</td>
</tr>
<tr>
<td>4.1.2. Small World Networks</td>
<td>61</td>
</tr>
<tr>
<td>4.1.3. Scale Free Networks</td>
<td>62</td>
</tr>
<tr>
<td>4.2. Statistical Properties of Complex Networks</td>
<td>64</td>
</tr>
<tr>
<td>4.2.1. Network Demographics</td>
<td>64</td>
</tr>
<tr>
<td>4.2.2. Network Density</td>
<td>64</td>
</tr>
</tbody>
</table>
# Table of Contents

4.2.3. Average Degree ........................................................................................................ 65
4.2.4. Scale Free Characteristic ..................................................................................... 65
4.2.5. Average Shortest Path Length ............................................................................. 66
4.2.6. Clustering Coefficient ......................................................................................... 66
4.2.7. Small World Network Characteristics .................................................................. 68
4.2.8. Percolation Theory ............................................................................................. 68
4.2.9. Diameter ................................................................................................................ 68
4.2.10. Community Structure ......................................................................................... 69

Chapter 5 - Defining System Structure and Characteristics .............................................. 70

5.1. Supply Chain ........................................................................................................... 70

5.1.1. Demand Planning in Supply Chains ................................................................. 73
5.1.2. Supply Planning in Supply Chains ........................................................................ 74
5.1.3. Supply Management in Supply Chains .............................................................. 75
5.1.4. Management of Other Resources in Supply Chains ........................................... 76

5.2. Triple-A Measures ................................................................................................... 77

5.2.1. Agility .................................................................................................................... 78
5.2.2. Alignment ............................................................................................................. 79
5.2.3. Adaptivity .............................................................................................................. 81

Chapter 6 – Formulation of an Enterprise System as a Complex Adaptive System .......... 82

6.1. Agency ...................................................................................................................... 84
6.2. Self-organization ....................................................................................................... 85
6.3. Emergence ............................................................................................................... 87
6.4. Network Connectivity ............................................................................................. 88
6.5. Dimensionality and Feedback ................................................................................. 90
6.6. Dynamic Process ..................................................................................................... 92
6.7. Adaptivity .................................................................................................................. 94
6.8. Rugged Landscape .......................................................................................................... 95
6.9. Co-evolution .................................................................................................................. 97
6.10. Quasi-equilibrium ......................................................................................................... 97
6.11. Non-linearity .............................................................................................................. 98
6.12. Non-random Future ...................................................................................................... 100
6.13. Difficulty to Determine Boundaries ............................................................................ 101

Chapter 7 – Modeling of CMEE ......................................................................................... 103
7.1. Agent Based Simulation Modeling .................................................................................. 109
7.2. Agent ............................................................................................................................. 110
7.3. Agent Attributes .......................................................................................................... 111
    7.3.1. Knowledge and Belief ............................................................................................ 111
    7.3.2. Inference ................................................................................................................. 111
    7.3.3. Social Relationship ................................................................................................. 112
    7.3.4. Goals ...................................................................................................................... 112
    7.3.5. Planning .................................................................................................................. 113
    7.3.6. Language ............................................................................................................... 113
    7.3.7. Knowledge Representation .................................................................................... 114
7.4. Environment ................................................................................................................. 114

Chapter 8 - Simulation Framework ...................................................................................... 115
8.1. Multi-Agent Framework of CMEE .................................................................................. 118
8.2. Agent Based Simulation Model of CMEE ..................................................................... 118
8.3. Agents of CMEE .......................................................................................................... 120
8.4. Agent Ecosystem ......................................................................................................... 128
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.5.</td>
<td>Network Structure of CMEE</td>
<td>130</td>
</tr>
<tr>
<td>9.1.</td>
<td>Controllable and Uncontrollable Variables</td>
<td>133</td>
</tr>
<tr>
<td>9.2.</td>
<td>State Variables / Attributes</td>
<td>133</td>
</tr>
<tr>
<td>9.2.1.</td>
<td>List of Global Variables</td>
<td>135</td>
</tr>
<tr>
<td>9.2.2.</td>
<td>List of Supplier Variables</td>
<td>137</td>
</tr>
<tr>
<td>9.2.3.</td>
<td>List of Customer Variables</td>
<td>138</td>
</tr>
<tr>
<td>9.3.</td>
<td>Process Overview and Scheduling</td>
<td>139</td>
</tr>
<tr>
<td>9.4.</td>
<td>Design Concepts</td>
<td>140</td>
</tr>
<tr>
<td>9.4.1.</td>
<td>Emergence</td>
<td>140</td>
</tr>
<tr>
<td>9.4.2.</td>
<td>Adaptation</td>
<td>141</td>
</tr>
<tr>
<td>9.4.3.</td>
<td>Fitness</td>
<td>142</td>
</tr>
<tr>
<td>9.4.4.</td>
<td>Prediction</td>
<td>144</td>
</tr>
<tr>
<td>9.4.5.</td>
<td>Sensing</td>
<td>145</td>
</tr>
<tr>
<td>9.4.6.</td>
<td>Stochasticity</td>
<td>146</td>
</tr>
<tr>
<td>9.4.7.</td>
<td>Collectives</td>
<td>147</td>
</tr>
<tr>
<td>9.4.8.</td>
<td>Observation</td>
<td>148</td>
</tr>
<tr>
<td>9.5.</td>
<td>Initialization</td>
<td>148</td>
</tr>
<tr>
<td>9.6.</td>
<td>Input</td>
<td>149</td>
</tr>
<tr>
<td>9.7.</td>
<td>Submodels</td>
<td>150</td>
</tr>
<tr>
<td>9.8.</td>
<td>Interface</td>
<td>150</td>
</tr>
<tr>
<td>9.9.</td>
<td>Performance Measures</td>
<td>150</td>
</tr>
<tr>
<td>9.10.</td>
<td>ABM Network Analysis</td>
<td>154</td>
</tr>
<tr>
<td>9.10.1.</td>
<td>Network Demographics of CMEE</td>
<td>154</td>
</tr>
<tr>
<td>9.10.2.</td>
<td>Network Density of CMEE</td>
<td>155</td>
</tr>
</tbody>
</table>
# Table of Contents

9.10.3. Average Degree of CMEE ........................................................................................................... 155

9.10.4. Average Shortest Path Length of CMEE ................................................................................... 156

9.10.5. Clustering Coefficient of CMEE ............................................................................................... 156

9.10.6. Small World Network Characteristics of CMEE ....................................................................... 157

9.10.7. Percolation Theory of CMEE ................................................................................................... 157

9.10.8. Diameter of CMEE .................................................................................................................... 158

Chapter 10 - Testing and Results ........................................................................................................... 159

10.1. Testing .......................................................................................................................................... 160


10.1.2. Black Box Testing ..................................................................................................................... 162

10.1.3. White Box Testing ..................................................................................................................... 163

10.2. Results of Statistical Analysis ....................................................................................................... 164

10.3. Results of ABM Network Analysis ............................................................................................... 188

10.3.1. Network Density ....................................................................................................................... 188

10.3.2. Average Degree ....................................................................................................................... 188

10.3.3. Average Shortest Path Length ................................................................................................. 189

10.3.4. Clustering Coefficient ............................................................................................................. 189

10.3.5. Small World Network Characteristics ..................................................................................... 189

10.3.6. Diameter .................................................................................................................................... 190

Chapter 11 - Conclusion and Future Work ........................................................................................... 192

11.1. Summary of Findings .................................................................................................................... 194

11.2. Future Work .................................................................................................................................. 200

References ............................................................................................................................................. 203

Appendix A: Submodels ......................................................................................................................... 213

A.1. Observer Submodels ....................................................................................................................... 213
A.2. General Agent Submodels (general to any agent type) .................................................. 228
A.3. Supplier Submodels ........................................................................................................ 229
A.4. Customer Submodels ...................................................................................................... 234
Appendix B: Interface ............................................................................................................. 235
B.1. Buttons ............................................................................................................................. 235
B.2. Sliders ............................................................................................................................... 235
B.3. Switches ............................................................................................................................. 236
B.4. Input ................................................................................................................................... 237
B.5. Monitor ............................................................................................................................... 237
List of Figures

Figure 1.1 Analogy between (a) biological systems and (b) enterprise systems .................. 20

Figure 8.1 Agent Based Simulation of CMEE ....................................................................... 119

Figure 10.1 Number of customer agents versus simulation clock ........................................ 166
Figure 10.2 Number of supplier agents versus simulation clock ........................................ 166
Figure 10.3 Market attributes versus environmental conditions ......................................... 167
Figure 10.4 Number of suppliers versus reproduction threshold for different environmental conditions .............................................................. 169
Figure 10.5 Adaptivity trends of neutral, adaptive and resistant suppliers ............................. 174
Figure 10.6 C.I. on mean adaptation size for neutral, adaptive and resistant suppliers .......... 174
Figure 10.7 One way ANOVA of adaptation size of adaptive suppliers ................................. 178
Figure 10.8 One way ANOVA of neutral size of neutral suppliers ..................................... 179
Figure 10.9 One way ANOVA of resistance size of resistant suppliers ................................. 179
Figure 10.10 Multiple pairwise comparisons of adaptive agents .......................................... 184
Figure 10.11 Multiple pairwise comparisons of resistant agents ......................................... 185
Figure 10.12 Multiple pairwise comparisons of neutral agents ........................................... 185
Figure 10.13 The number of cooperative versus greedy suppliers ....................................... 187
Figure 10.14 C.I. on mean cooperation size ......................................................................... 187
List of Tables

Table 2.1 Computer Components ................................................................. 35

Table 7.1 Selected types of simulation and how they operate (Allen 2011) ..................... 106

Table 9.1 Controllable and uncontrollable variables .................................................... 134

Table 10.1 The factors and factor levels ...................................................................... 170
Table 10.2 The standard ANOVA table of adaptation size of adaptive suppliers ............... 170
Table 10.3 Results for design of experiments study ...................................................... 171
Table 10.4 The standard ANOVA table of adaptation size of adaptive suppliers ............... 177
Table 10.5 The standard ANOVA table of adaptation size of neutral suppliers .................. 177
Table 10.6 The standard ANOVA table of adaptation size of resistant suppliers ............... 178
Table 10.7 Multiple pairwise comparisons of adaptive agents performed at the 0.01 level ...... 182
Table 10.8 Multiple pairwise comparisons of resistant agents performed at the 0.01 level ...... 183
Table 10.9 Multiple pairwise comparisons of neutral agents performed at the 0.01 level ...... 184
Chapter 1

Introduction

Today’s business environment is a rapidly changing ecosystem. In this ecosystem, environmental disturbances force enterprises to quickly adapt to changes. This adaptation is required to secure or improve the current position of an enterprise in the market. Enterprises try different approaches to survive these environmental disturbances and develop different strategies to anticipate the forthcoming changes. An unprepared enterprise operating in this environment will be extinct sooner or later when faced with rapid changes in the business ecosystem; however it will survive if it is agile enough to transform itself into an enterprise that fits into the ecosystem and aligns its goals with those of suppliers and customers.

Similarly, the continuation of a product or system in a market is as important as its innovation and realization (Cavinato 1992; Ellram 1993; Lee 1993). Enterprises try different approaches to sustain their products and survive environmental disturbances. They develop different strategies to anticipate and cope with forthcoming changes. Consequently, many enterprises have spent increasing amounts of time, money, and effort to predict system changes and control system behavior.
However, the efforts to manage enterprise systems have often led to frustration and helplessness. Managers have struggled with the prediction and control of dynamic and complex nature of enterprise ecosystems. For instance, rapid changes that occur at different parts of the ecosystem are often outside the purview and control of a single enterprise.

Therefore, we study the influence and effect of decisions and actions of enterprises in a dynamic, rapidly changing and uncertain ecosystem. We build a model that covers the concept of self-organization from the biological ecosystems (Ito and Gunji 1994; Sneppen, Bak et al. 1995; Pascual, Roy et al. 2002) and the reconfigurability concepts from manufacturing systems (Koren, Heisel et al. 1999; Zhang, Chan et al. 2002; Setchi and Lagos 2004; Su and Chen 2004; Tang and Qiu 2004; Wang, Hou et al. 2005; ElMaraghy 2006) to enable enterprises to anticipate and synchronize with the changing trends in emerging computer manufacturing environment.

To be more specific, in our research, we address issues of computer manufacturing collaboration, customer and supply network and we call it computer manufacturing enterprise ecosystem (CMEE). We treat CMEE as a complex adaptive supply network (CASN) and investigate the evolution of topological structures and dynamic behaviors of the CMEE via an agent-based simulation model. In our research, we take the agility, alignment and adaptivity measures, which we called triple-A measures, as the key components of successful survival in the business ecosystem. The agent-based simulation model that is developed in this research contains triple-A measures implemented into customer and supplier agents. Each agent uses different algorithms and rules to exhibit intelligent behavior and to optimize its choices. The simulation model is
analyzed using well-established methods (Pritsker 1979; Kelton 1998; Law 2007) and network science theory (Wasserman and Faust 1994; Barabási 2002).

In the rest of this chapter we provide an overview of current business environment and our approach to addressed issues. It continues with our research objectives and research motivation. Then we explain the research statement and provide the solution and state the contribution of our work.

1.1. Overview

Today’s business environment is largely characterized by four words: change, speed, uncertainty and connectivity. An unprepared enterprise operating in this environment will be extinct sooner or later when faced with rapid changes in the business ecosystem; however it will survive if it is agile enough to transform itself into an enterprise that fits into the ecosystem. Environmental disturbances can come from three primary sources: (1) customer demand for individualized and customized products at lower costs; (2) predictable changes created by such factors as technology advances, social trends, and environmental concerns; and (3) unpredictable changes such as natural calamities, war threats, and technological breakthroughs.

John Scully, the former chairman and CEO of Apple Computer (NGM Project Office 1997) stated that “The companies that succeed in the long run will be those with the courage to
quickly transform their businesses to capitalize on change.” Carly Fiorina, former CEO of HP (Fiorina 2003) said “The company that thinks it’s done is done.” Darwin [referring to his theory of evolution] said “it’s not the most intelligent or strongest that survives, but those that adapt the most rapidly to change.” “It’s not about fixing a company and stopping it, but it’s about a company being able to adapt.” Bill Gates wrote in his book (Gates 2001) that the 1980’s were all about computing, the 1990’s were all about bandwidth, and this decade will be all about change and the increased velocity with which everything will be changing.

We envision that the paradigm of agile, adaptive, aligned enterprises is a tactical solution to realize a successful enterprise. We define a triple-A enterprise as “a virtual enterprise that can self-organize itself to efficiently operate in an environment marked by unforeseen and unforeseeable technological, social, and global changes.” This paradigm draws its inspiration from biological systems as shown in Figure 1.1. It imitates the survivability mechanism of biological systems. The proposed paradigm will enable enterprises to address the two key issues identified by CEOs of top companies at their meeting organized by the eBusiness Research Center (Pal and Pantaleo 2005):

Figure 1.1 Analogy between (a) biological systems and (b) enterprise systems
(1) Anticipate the forthcoming changes and rapidly transform the organizational structure to thrive in on the changes – in contrast to the traditional enterprises which respond to the changes that occurred in past.

(2) Synchronize the infrastructure changes with the anticipated environmental changes – just aligning infrastructure to currently changing needs is not sufficient.

1.2. Research Objectives

The ultimate goal of this dissertation is to provide computer manufacturing enterprises with methodologies and tools to transform and optimally reconfigure themselves to competitively operate in a rapidly changing business environment. The objective of the dissertation is to create representations, metrics, algorithms, and methods to realize robust reconfigurable CMEE that can convert a change in the business environment into a value proposition.

Moreover, the proposed work will show that computer manufacturing enterprises exhibit self-organizing behavior – in the sense, they operate adaptively, evolve continuously, and even transform into new units over a long period of time. Further the proposed work intends to help policy makers and business holders to understand the dynamics of the formation, adaptation and evolution of CMEE that set the rules of survival and growth in the business environment.

Therefore, the main objective of the dissertation is to formulate a model of computer manufacturing enterprises that adapt foreseen and unforeseen changes quickly, synchronize
shared goals and objectives with other enterprises, and able to give decisions rapidly and efficiently as soon as the environment surrounding them changes.

1.3. Motivation

In a study of 1998 (Swaminathan, Smith et al. 1998), Swaminathan found that the decision-making process in the supply chain for a computer manufacturer was centralized to a great extent; few suppliers were extremely important whereas others were mainly controlled by the manufacturer, and a major part of the supply chain was owned by the manufacturer.

However, today’s supply networks exist in a diverse, heterogeneous, and hierarchical ecosystem that evolves in an environment sensitive to initial conditions or to small perturbations. Therefore, a traditional supply network with a tree-like or hierarchical structure lacks the main survivability, flexibility, scalability and adaptivity properties.

Therefore currently, there is a lot of vagueness and confusion about how to become a successful, long-term enterprise both in academia and industry. Awareness of complex adaptive system and their dynamic, emerging and self-organizing structure make enterprises realize that they are faced with rapid changes in the business ecosystem and in order to survive they need the right tools to be able to transform into enterprises that fit into the ecosystem and align their goals with those of suppliers and customers. That leads enterprises to try different approaches to survive environmental disturbances and develop different strategies to anticipate the forthcoming changes.
Lee (Lee 2004) proposed that a supply chain that consists of adaptivity, alignment and agility measures (triple-A supply chain) is key to an enterprise’s success in today’s business markets. However, so far there are no tools and techniques, which can be readily used to analyze and study the performance of a triple-A supply chain.

The confusion that prevails today around adaptivity, alignment and agility must be eliminated by proper scientific investigation into the concept of triple-A supply chain. A formal definition of triple-A supply chain measures and metrics to quantify them are required. In order to understand and implement triple-A supply chain measures, a new set of methodologies are required. This forms the fundamental motivation for this dissertation.

1.4. Problem Statement

One of the challenging, rapidly changing but highly profitable markets is computer manufacturing enterprise ecosystem (CMEE). CMEE is the integration of computer manufacturing enterprises, their suppliers, related industries, customers, products, services, and information that add value to computer manufacturing enterprise ecosystem. In this ecosystem it is important to balance efforts of customer satisfaction, manufacturing and inventory costs and provide a timely good response to environmental disturbances. This inherent complexity and necessity to search for right tools to optimize market survival lead us to investigate the underlying structure of this ecosystem.
In this dissertation we propose that computer manufacturing enterprise ecosystem (CMEE) is a complex adaptive system. Therefore our goal is to investigate the evolution and self-organization of computer manufacturing enterprise ecosystem and to formulate an integrated model that covers the concept of self-organization from the biological ecosystems (Ito and Gunji 1994; Sneppen, Bak et al. 1995; Pascual, Roy et al. 2002) and the reconfigurability concepts from manufacturing systems (Koren, Heisel et al. 1999; Zhang, Chan et al. 2002; Setchi and Lagos 2004; Su and Chen 2004; Tang and Qiu 2004; Wang, Hou et al. 2005; ElMaraghy 2006) for building agile, adaptive, and aligned computer manufacturing enterprises. Our work provides computer manufacturing enterprises with agent based and network theoretic methods and tools to optimally reconfigure themselves to thrive in the opportunities created by unexpected changes and scenarios.

To summarize, we propose that:

- Computer manufacturing enterprise ecosystem is a complex adaptive system.

- Paradigm of reconfigurable enterprises is a tactical solution to realize an agile, adaptive and aligned computer manufacturing enterprise ecosystem.

### 1.5. Solution

Computer manufacturing enterprises evolve in an uncertain quickly changing ecosystem. This ecosystem consists of competitors, collaborators, suppliers, customers, interactions between those and environmental effects. In this dynamic ecosystem, every step an enterprise
take involves certain risks, and risks are the product of hazards and vulnerability. An enterprise continuously adapts and synchronizes itself throughout internal and external linkages to overcome the consequences of inappropriately managed risk. Therefore, a full understanding of the computer manufacturing ecosystem is essential for taking calculated risks and making sensible investment decisions. In the light of this mind set, we formulate computer manufacturing ecosystem as a complex adaptive system. We investigate the existing inherent complexity of this ecosystem utilizing the agent based simulation tools with which we strengthen by adding the network theoretic metrics. This approach makes it possible to observe emergence and evolution of computer manufacturing market, and its underlying structure.

In this dissertation, our main goal is to provide the necessary tools to computer manufacturing enterprises for survival in fast paced, altering, complex business market environment. In order to achieve our goal, we carried out a dedicated study on previous researches and current structure of computer manufacturing enterprise ecosystem. As a result of our studies, we recognized computer manufacturing enterprise ecosystem as a complex adaptive system. Therefore, in the light of recent studies, we decided to use agent based simulation approach and network theoretic approaches to provide desired tools. In our research, we mainly focus on triple-A supply chain (Lee 2004) features, namely; adaptivity, alignment and agility.
1.6. Contribution

Based on the above theory, we plan to derive and investigate adaptation mechanisms in CMEE to better understand their possible evolutionary paths. More specifically we plan to deliver the following: (1) methods to study evolution and adaptation of CMEE, (2) a comprehensive model for understanding CMEE, (3) a comprehensive study on how uncertain and rapidly changing business environment effects the CMEE, (4) develop an agent based simulation by considering emergence and self-organization in the CMEE, (5) develop a network model of the resultant simulation, and (6) perform network analysis of the resultant model.

The construction of algorithms and software that simulate aspects of CMEE will contribute to the understanding and building of agile, adaptive, and aligned (triple-A) CMEE. It will also help to find enterprise architectures which bestow the right balance between being rigid and being conducive to modularization and self-organization by using agent based approach.

This research work also strongly supports the global compatibility and competitiveness of US Industries. We will show that the use of complex adaptive systems and network theory allows timely diagnosis and detections of abnormalities in the operations of a CMEE. As a result computer manufacturing enterprises, which are agile and robust to uncertainties and rapid changes, function well in a dynamic business environment. Therefore building a triple-A CMEE establishes an agile and robust profile for US computer manufacturing companies that are well prepared to global changes.
The project offers important and valuable contributions and benefits to both policy makers and companies. For policy makers the outcomes of the research help to (1) formulate the next-generation reconfigurable computer manufacturing enterprises that meet the rapidly changing market and industry requirements better; (2) gain better understanding of evolution and organization of CMEE; and (3) identify the requirements of an agile and robust CME. For companies the research helps to (1) satisfy customer demand for individualized and customized products at lower costs; (2) prepare for predictable changes created by factors such as technology advancements, social trends, and environmental concerns; (3) guard against unpredictable changes such as natural calamities, war threats, and technological breakthroughs; (4) offer high-quality mass customized products at low cost with utmost time and delivery flexibility; and (5) secure a competitive edge.
Chapter 2

Literature Review

In a rapidly changing computer manufacturing ecosystem it becomes more difficult to diagnose system deficiencies (lack of networking, lack of dynamism, lack of control, etc.), predict forthcoming future (customer demand, technology advancements, social trends, war treats, etc.) and organize the system properly (adaptation and synchronization). The process of deficiency realization, decision making, transformation and adaptation to rapidly changing markets raises some research questions such as how to approach the changes and which techniques to use during the transition period.

Hence scientists search for a model that reflects this underlying complexity and dynamism. Most scientists choose to develop or use some kind of computer simulation to capture and understand the complexity of enterprise systems. Also there is a dearth of research that explains the processes that may govern the emergence, growth, and evolution of enterprise system topologies over time (Harland, Lamming et al. 2001). Kaihara (2003) proposed a supply chain management (SCM) based on market-oriented programming that computes multi-agent behavior and implements the distributed computation as a market price system. Kim (2009) modeled a supply network as a complex adaptive system (CAS) in which firms or agents interact with one another.
and adapt themselves. He used an agent-based social simulation (ABSS) model, a research method of simulating social systems under the CAS paradigm, to observe emergent outcomes.

Just as Kim, many scientists recognized supply networks as CASs. In a 2009 study, Pathak, et al. (2009) investigated the dynamics of a complex adaptive supply network, to understand the stability of the structural evolution of a supply network and supplier population emergence. Their analysis revealed that the type of environment a supply network evolves in appears to be a major factor in determining critical timing of structural changes during the evolution of a complex adaptive supply network. Li, et al. (2009) conducted a multi-agent simulation study on the evolution of complex adaptive supply networks, and reported that the supply network emerges and evolves from the firms’ dynamic interaction under the dynamic environment. In the article of Nair, et al. (2009), the authors examined how the firms embedded in supply networks engage in decision making over time. They build a simulation model using cellular automata to investigate the dynamic evolution and defection among complex adaptive supply network agents. From a policy maker/strategist’s point-of-view, the impact of these dynamic forces on the growth and evolution structure of networks is interesting and challenging (Choi and Hong 2002). Because, policy makers often create laws and regulations with a vision that laws and regulations will lead to the evolution of certain types of desirable enterprise systems.

However, unanticipated changes in business environment faced by most enterprise systems often negate such intentions. An example of unanticipated impact of a policy is HIPAA (T. Y. WebMD Corporation 2004). In 1996, the U.S. Congress passed the Healthcare Insurance Portability and Accountability Act (HIPAA) that mandated adoption of a set of regulations.
HIPAA is related to standards and requirements for the electronic submission of health information. The aim of this act is to eliminate the intermediary clearinghouses, which convert the diverse forms from one structure to another. However, during this transformation the fact that most local providers are not ready to implement those changes is completely ignored. The local providers, which faced extinction in this rapidly changing environment, stepped into an agreement with intermediary clearinghouses for their survival.

Thus instead of eliminating the clearinghouses, the position of intermediaries in the system was strengthened (T. Y. WebMD Corporation 2004). There are other examples where collaborations in an enterprise system have dramatically changed the “rules” of the network. The classic example of this is the effect of Microsoft and Intel (“WinTel”) on the IBM personal computer (PC) supply network, where the suppliers went from a low-power position to dominating the environment (Fine 1998; Mazzucato 2002). A more recent example is the logic-products industry (Lewis 2000) where it was predicted that given regulations and state of the industry, logic-products (used for designing circuits) would die out and that this network layer would vanish. What has happened is the opposite: with the adoption of new strategies to reduce logic-device prices, the industry has never been healthier.

Those examples show that an enterprise, more specifically a computer manufacturing enterprise, should be able to handle successfully both predictable and unforeseeable changes. This is well expressed in Ashby’s Law (Ashby 1964; Tharumarajah 2003); the variety of changes (disturbances) that an enterprise can potentially withstand should be equal to or greater than the variety of changes induced by its business environment. Of course this is not a practical
possibility. Alternatively we believe that the issue can be addressed if computer manufacturing enterprises have three additional properties: adaptability, agility and alignment. Adaptability is a concept borrowed from biological systems, which takes place over a long time, whereby an enterprise becomes better suited to the business environment. Agility refers to the ability to perform different functionalities almost with equal dexterity. Finally, alignment refers to synchronization of shared goals and objectives of a set of enterprises in a business environment. These three properties are named as triple-A measures. In a business environment, a triple-A enterprise is likely to encounter both familiar as well as unfamiliar situations. Though, in order to possess the ability to deal with rapidly changing uncertain environmental rules, the triple-A enterprise should be able to self-diagnose a change and take actions that modify the operating norms if necessary.

However, we realize that there are several challenges in building triple-A enterprises. A triple-A enterprise, with its boundaries often beyond a single enterprise, would invite additional spatial and temporal complexities. Triple-A enterprises are expected to come into the existence quickly, operate without undue compromise on short-term efficiencies, and dissolve as soon as the environment surrounding them changes. Moreover, although the structures and collaboration mechanisms of an enterprise ecosystem are static in the short term, they evolve in the long run. The optimal network structure and collaboration mechanism of enterprise systems, which take researchers’ many efforts based on the assumption of static structure, may become invalid as the systems evolve. To better facilitate the management of enterprise systems, we need to understand more about the dynamic behaviors of the enterprises and enterprise ecosystem. These practical
realities give rise to several issues that need to be addressed in this dissertation. A sample of key issues is listed below:

- How do different enterprises form an enterprise ecosystem? (Emergence and Topology)
- How does the enterprise ecosystem evolve over time? (Evolution)
- What enterprise architectures would bestow the right balance between being rigid and being conducive to connectivity, dynamism and self-organization? (Connectivity, Dynamism and Self-organization)
- What mechanisms would allow timely detections of system deficiencies and abnormalities in the operations of an enterprise system? (Agility)
- What operational features ensure that enterprises function well in a dynamic business environment? (Agility, Adaptivity, Alignment)

To answer these issues, we need to understand the dynamics of the formation, evolution and adaptation of enterprise ecosystem. We have seen considerable progress in reconfigurable computing (Estrin 1960; Yigit and Allahverdi 2003; Bobda 2005; Fu and Compton 2005), reconfigurable products (Yigit, Ulsoy et al. 2002; Abdi and Labib 2004), reconfigurable assembly (Guisti, Santochi et al. 1994; Yu, Yin et al. 2003; Lohse, Hirani et al. 2005), reconfigurable material handling (Ho and Ranky 1994; Kolluru, Valavanis et al. 2000; Babiceanu, Chen et al. 2004), reconfigurable robots (Bojinov, Casal et al. 2002; Salemi and Shen 2004; Shen, Krivokon et al. 2006), reconfigurable manufacturing cells (Zhu, Sheng et al. 2004; Endsley, Almeida et al. 2006), and reconfigurable manufacturing systems (Koren, Heisel et al. 2005).
Chapter 2

1999; Zhang, Chan et al. 2002; Setchi and Lagos 2004; Su and Chen 2004; Tang and Qiu 2004; ElMaraghy 2005; Wang, Hou et al. 2005). Taking lead from these areas, we plan to investigate the dynamics of computer manufacturing enterprise ecosystem by using concept of reconfigurability. We propose that a computer manufacturing enterprise can reconfigure and defend itself against unexpected disturbances by understanding the complex adaptive systems and triple-A measures.

So far we provided the background of our study, reviewed what is known and done until now; at this point we present a picture narrowing in on computer manufacturing enterprise systems.

2.1. Introduction to Computers and Computer Components

Here an introduction to computers and computer components is provided in order to better understand the complexity involved in computer manufacturing industry. Computers come in all types and sizes. There are primarily two main sizes of computers: portable computers and desktop computers. The portable computers come in various sizes and are referred to as laptops, notebooks, and hand-held computers. These generally denote different sizes, the laptop being the largest, and the hand-held is the smallest size.

Computer hardware is the physical part of the computer including the digital circuits inside the computer as opposed to the software that carry out the computing instructions. The hardware of a computer is unlikely to change frequently unless due to the crash or for upgrading them. Hardware comprises all of the physical part of the computer such as
Monitor, CPU, motherboard, ram, CD-Rom, printer, scanner, hard disk, flash drive (AKA pen drive), processor, PCI buses, floppy disk, power supply, VGA card, sound card, network interface card, peripherals, joystick, mouse, keyboard, foot pedal, computer fan, camera, headset and others.

On the other hand, software is a logical part of a computer and is used to carry out the instructions, storing, executing, and developing other software programs. A typical PC consists of a case or chassis in the desktop or tower case, and the components that are shown in Table 2.1.

**CPU (Central Processing Unit):** CPU or central processing unit relates to processor. The performance of the computer is determined by the CPU chip (processor speed) and the other computer circuitry. Currently, the most popular chip brand is the Pentium chip (processor) and it has close competitors available in the market such as AMD and Motorola. The clock speed becomes the most important factor in determining the performance of a computer. The motherboard contains the hardware circuitry and connections that allow the different hardware components of the PC to interact and communicate with each other. Most computer software is being developed for the latest processors so it would be difficult to use them with the older systems.
<table>
<thead>
<tr>
<th>Motherboard</th>
<th>PC Buses</th>
<th>Media</th>
</tr>
</thead>
<tbody>
<tr>
<td>• CPU</td>
<td>• PCI</td>
<td>• CD-Rom</td>
</tr>
<tr>
<td>• Computer Fan</td>
<td>• USB</td>
<td>• DVD-Rom</td>
</tr>
<tr>
<td>• RAM</td>
<td>• Hyper-transport</td>
<td>• Combo box</td>
</tr>
<tr>
<td>• BIOS</td>
<td>• AGP</td>
<td>• Joystick</td>
</tr>
<tr>
<td>• Digital Circuitry</td>
<td>• ISA</td>
<td>• BD-Rom drive</td>
</tr>
<tr>
<td>• Computer Fan</td>
<td>• EISA</td>
<td></td>
</tr>
<tr>
<td>• PCI Slots</td>
<td>• VLB</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Internal storage</th>
<th>Other Peripheral Devices</th>
<th>Output Devices</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Hard disk (ATA &amp;</td>
<td>• Mouse</td>
<td>• Printer</td>
</tr>
<tr>
<td>SATA)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Data array controller</td>
<td>• Modems</td>
<td>• Scanner</td>
</tr>
<tr>
<td>• Floppy disk</td>
<td>• Digital Camera</td>
<td>• Monitor</td>
</tr>
<tr>
<td></td>
<td>• Sound, Video Cards</td>
<td>• Speakers</td>
</tr>
<tr>
<td></td>
<td>• Printer</td>
<td>• Headset</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Video output devices</td>
</tr>
</tbody>
</table>
**Hard Disk Drives:** Disk drive is the mechanism to run the disks. All disks need a drive to get the information, read it and put it back to the disks. Hard disk is used to store the data permanently. Often the terms disk and drive used to describe the same thing but it should be clear that a disk is a storage device.

**Modem:** A modem is used for the modulation and demodulation of data that is transferred through the modem and the telephone lines. The modem translates data from digital to analog and from analog to digital. It turns the digital data of a computer into modulated electrical signals in the voice frequency range of a telephone channel. These signals are transmitted over telephone lines and demodulated by another modem at the receiver side to recover the digital data. Modems are measured by the speed which is called baud rate. The typical baud rate is 56Kb.

**Keyboard:** The keyboard is used to type something or input information to the computer. There are different designs and models of the keyboards in the market. The most common layout of the keyboard is QWERTY layout. A standard keyboard has 101 keys and embedded keys.

**Video Cards:** Video cards allow computer to display video, graphics and animation. Some video cards allow computers to display television. A video card with a digital video camera allows users to produce live video. A high speed broadband internet connection is required to watch the videos on the Internet.
**Network Cards:** Network interface cards (NIC) allow PCs to connect with each other and communicate. Every network computer is required to have a NIC card. NIC cards are required both in wired and wireless networking.

**Cables:** There are two broad types of cables internal cables, which are embedded on the motherboard circuit that performs the communication between the devices and CPU. The other types of the cables are the network cables like coaxial cable, CAT 5, Ethernet cables. These cables are used for the communication purposes between the devices or computers.

**Memory:** Memory is the one of the important piece of the hardware. Sometimes memory chip resident memory is confused with the hard disk memory. Occasionally unallocated space of the hard disk is used as virtual memory also known as page file. This type of memory is used temporarily when the actual memory is inadequate to perform a specific task.

**RAM (Random Access Memory):** RAM is a memory that is being used by the computer to store information temporarily. For example when a computational task is performed on some applications that work is temporarily stored in the RAM. The more the RAM the faster the computer works. Modern PC requires at least 64MB RAM. RAM is in the form of a chip and different vendors have developed the RAM of different capacities.

**Mouse:** Every modern computer requires a mouse for faster operations. Generally a mouse has two buttons left and right to perform different functions. One type of mouse has a round
ball under the bottom. Another type of mouse uses an optical system to track the movement of the mouse.

**Monitors:** The monitor is used to display the information on the screen. All the activities of a computer, functions and tasks are seen on the computer screen and this is called outputting information. Monitors come in many sizes and shapes, monochrome or full colors. Today most computers use LCD screens. LCD screen is light weight and consumes less power as compared to the monitors.

**Printers:** The printer takes the information from the PC and transfers it to the paper of different sizes, which are placed in the printer device. There are three basic types of a printer such as dot matrix, inkjet and laser.

**Scanners:** Scanners allow you to transfer pictures and photographs to your computer. A scanner is used to scan the images and pictures. You can then send the image to someone, modify it or take a print out of it. With optical character recognition software one can convert a printed text into an editable text.

**Digital Camera:** One can take the digital photographs with the digital cameras. The images are stored on the memory chip of the digital cameras and one can transfer them to a computer with the USB drive.
**Case:** Case or casing covers the computer’s circuitry. There are two types of casings: desktop and tower casing. There is room inside the casing to add or remove components. Cases come in many sizes like desktop, mini, midi and tower. There are some additional empty slots inside the cases such as IDE, USB, ASI, PCI and firewire slots.

**Cards:** Cards are the hardware components that are added to the computer to increase their functionalities and capabilities. Sound cards produce the sound like music and voice. The older sound cards were produced in 8 and 16 bit options. Today the industry standard changed to 32 bit sound cards. Color cards allow computers to produce colors. Initially there were 2, 4 and then 16 bit color cards. The main types of the graphic cards are EGA, VGA and SGA. Today the 32 bit cards are the standard to display almost billions of the colors on the monitor.

### 2.2. Nature of Computer Manufacturing Industry

In the previous sections, we presented a short overview of the problem faced by enterprise systems. The main concentration of this dissertation is the computer manufacturing enterprise systems that are a specialized subset of overall enterprise systems. So let’s give a brief description of this industry sector.

The computer and electronic product manufacturing industry produces computers, computer peripherals, communications equipment, and similar electronic products. These products are used in homes and businesses, as well as in government and military establishments. In
addition, many electronics products or components are incorporated into other industries’ products, such as cars, toys, and appliances.

Technological innovation characterizes this industry more than most others and, in fact, drives much of the industry's production. This unusually rapid pace of innovation and technological advancement requires an extensive research and development (R&D). Likewise, the importance of promoting and selling the products manufactured by various segments of the industry requires knowledgeable marketing. American companies in this industry manufacture and assemble many products abroad to take advantage of lower production costs and favorable regulatory environments.

Although some of the enterprises in this industry are very large, most are relatively small. The tradition of innovation in the industry explains the origins of many small enterprises. Some enterprises are involved in design or R&D, whereas others may simply manufacture components, such as computer chips, under contract for others. Often, new companies are set up to develop an associated product. Once developed, the enterprise licenses a production enterprise to manufacture the product, which is then sold by the original enterprise.

Although electronic products can be quite sophisticated, production methods are often similar, making it possible for a single enterprise to manufacture many different electronic products or components with a relatively small investment. Investors often are willing to put their money behind new enterprises in this industry because of historically large paybacks.
Rapid technological innovation is an industry trademark. Companies invest in information systems to support new product development, increase manufacturing efficiency, and manage product distribution. Given the short product life of many new products, manufacturing processes and technologies are designed to be flexible and easy to change.

2.3. Organization of Computer Manufacturing Industry

The computer and electronic product manufacturing industry has many sectors. Companies in the industry are generally classified by what they sell.

Computer and peripheral manufacturing is made up of companies that make computers and related products, known as peripherals. Most computers are built by a small number of well-known brands, but there are also many small companies that sell their products locally or on the Internet. Computers are made up of components, such as motherboards, central processing units, graphics cards, hard disk drives, and power supplies. Many of these products are purchased from other companies and assembled as part of the computer. As a result, many finished computers are simply the combination of a number of other products.

Other firms in this industry sector produce computer peripherals, which are devices that can be used with computers, such as keyboards, mice, printers, and scanners. Other peripherals, generally known as internal peripherals, are physically installed in the computer's case. These include hard disk drives, networking cards, modems, sound cards, and disk drives. Many
internal peripherals are prepackaged as part of a computer, although almost all of them can be installed by a technician or experienced computer owner.

Semiconductor and other electronic component manufacturers produce integrated circuits, or computer microchips, which power a wide range of electronic products. They also produce other electronic components, such as resistors and capacitors, as well as printed circuit boards. Unlike most of the companies in this industry, these manufacturers start from basic materials such as silicon and copper and produce intermediate products that are only rarely sold directly to consumers. The exceptions to this rule include companies that produce central processing units and memory chips, although even these products are more likely to be pre-installed in a new computer.

The navigational, measuring, electro-medical, and control instruments manufacturing segment is a diverse group of companies that produce products mainly for industrial, military, and healthcare use. It also includes some consumer products, such as global positioning system (GPS) devices, as well as clocks and watches. This segment is one of the largest in the industry, mainly because its primary customers are the U.S. Department of Defense and the healthcare industry.

Manufacturing and reproducing magnetic and optical media is another segment of this industry. Firms in this segment produce blank CDs, DVDs, and audio and video tapes. They produce some of these blank media for sale to consumers, but most of the blank media is used to duplicate audio recordings, videos and movies, software, and the remaining media for
distribution to consumers and business users on a mass scale. Establishments in this segment are usually either subsidiaries of companies that create the software, movies, or recordings or are independent firms licensed by such companies as distributors.
Chapter 3

Complex Systems

Complex Systems is a new field of science studying how parts of a system give rise to the collective behaviors of the system, and how the system interacts with its environment. A complex system is the system composed of interconnected parts that as a whole exhibit one or more properties (behavior among the possible properties) not obvious from the properties of the individual parts.

- A complex system is a highly structured system, which shows structure with variations (Goldenfeld and Kadanoff 1999).

- A complex system is one whose evolution is very sensitive to initial conditions or to small perturbations, one in which the number of independent interacting components is large, or one in which there are multiple pathways by which the system can evolve (Whitesides and Ismagilov 1999).

- A complex system is one that by design or function or both is difficult to understand and verify (Weng, Bhalla et al. 1999).

- A complex system is one in which there are multiple interactions between many different components (Rind 1999).

- Complex systems are systems in process that constantly evolve and unfold over time (Arthur, Durlauf et al. 1997).
Social systems formed out of people, the brain formed out of neurons, molecules formed out of atoms, the weather formed out of air flows are all examples of complex systems. The field of complex systems cuts across all traditional disciplines of science, as well as engineering, management, and medicine. It focuses on certain questions about parts, wholes and relationships. These questions are relevant to all traditional fields.

A system’s complexity may be of one of two forms: disorganized complexity and organized complexity (Weaver 1948). In essence, disorganized complexity is a matter of a very large number of parts, and organized complexity is a matter of the subject system (quite possibly with only a limited number of parts) exhibiting emergent properties. Other examples of complex systems include ant colonies, human economies, climate, nervous systems, cells and living things, including human beings, as well as modern energy or telecommunication infrastructures. Indeed, many systems of interest to humans are complex systems.

Complex systems are studied by many areas of natural science, mathematics, and social science. Fields that specialize in the interdisciplinary study of complex systems include systems theory, complexity theory, systems ecology, and cybernetics. Also, during the recent years, complex systems have been approached from several perspectives. Especially the complex networks view has turned out to be extremely fruitful, revealing general principles applicable to a large number of systems ranging from the Internet to protein-protein interaction networks of living cells. In this approach, diverse systems are viewed as networks, so that the interacting elements such as
proteins are described by network vertices and their interactions by edges (links) connecting the vertices.

Complex systems concepts provide a new unity of approach to many different problems. These concepts originate from efforts to understand physical, biological and social systems. Examples of applications can be found in all fields and professions including science, medicine, engineering, management and education. It should be recognized that complex systems is an active field and the most exciting discoveries are yet to come.

3.1. Why Complex Systems?

The study of complex systems is about understanding indirect effects. Problems that are difficult to solve are often hard to understand because the causes and effects are not obviously related. Disturbing a complex system at one point may create its effect at some other point because the parts are interdependent. This has become more and more apparent in our efforts to solve societal problems or avoid ecological disasters caused by our own actions. The field of complex systems provides a number of sophisticated tools, some of them concepts that help us think about these systems, some of them analytical for studying these systems in greater depth, and some of them are computer based for describing, modeling or simulating these systems.
3.2. Three Approaches to the Study of Complex Systems

There are three interrelated approaches to the modern study of complex systems, (1) how interactions give rise to patterns of behavior, (2) understanding the ways of describing complex systems, and (3) the process of formation of complex systems through pattern formation and evolution.

3.3. Types of Complex Systems

3.3.1. Chaotic Systems

For a dynamical system to be classified as chaotic, most scientists will agree that it must have the following properties:

- It must be sensitive to initial conditions
- It must be topologically mixing
- Its periodic orbits must be dense

Sensitivity to initial conditions means that each point in such a system is arbitrarily closely approximated by other points with significantly different future trajectories. Thus, an arbitrarily small perturbation of the current trajectory may lead to significantly different future behavior.
3.3.2. Complex Adaptive Systems

Complex adaptive systems (CAS) are special cases of complex systems. They are complex in that they are diverse and made up of multiple interconnected elements and adaptive in that they have the capacity to change and learn from experience. Examples of complex adaptive systems include the stock market, social insect and ant colonies, the biosphere and the ecosystem, the brain and the immune system, the cell and the developing embryo, supply chains, manufacturing businesses and any human social group-based endeavor in a cultural and social system such as political parties or communities.

3.3.3. Nonlinear System

A nonlinear system is one whose behavior can't be expressed as a sum of the behaviors of its parts (or of their multiples.) In technical terms, the behavior of nonlinear systems is not subject to the principle of superposition. Linear systems are subject to superposition.

3.4. Features of Complex Systems

Complex systems have the following features.

**Boundaries are difficult to determine:** It can be difficult to determine the boundaries of a complex system. The decision is ultimately made by the observer.
**Complex systems may be open:** Complex systems are usually open systems, that is, they exist in a thermodynamic gradient and dissipate energy. In other words, complex systems are frequently far from energetic equilibrium: but despite this flux, there may be pattern stability.

**Complex systems may have a memory:** The history of a complex system may be important. Because complex systems are dynamical systems they change over time, and prior states may have an influence on present states. More formally, complex systems often exhibit hysteresis.

**Complex systems may be nested:** The components of a complex system may themselves be complex systems. For example, an economy is made up of organizations, which are made up of people, which are made up of cells, all of which are complex systems.

**Dynamic network of multiplicity:** As well as coupling rules, the dynamic network of a complex system is important. Small-world or scale-free networks which have many local interactions and a smaller number of inter-area connections are often employed. Natural complex systems often exhibit such topologies. In the human cortex for example, we see dense local connectivity and a few very long axon projections between regions inside the cortex and to other brain regions.

**May produce emergent phenomena:** Complex systems may exhibit behaviors that are emergent, which is to say that while the results may be deterministic, they may have properties that can only be studied at a higher level. For example, the termites in a mound have physiology, biochemistry and biological development that are at one level of analysis,
but their social behavior and mound building is a property that emerges from the collection of termites and needs to be analyzed at a different level.

**Relationships are non-linear:** In practical terms, this means a small perturbation may cause a large effect (butterfly effect), a proportional effect, or even no effect at all. In linear systems, effect is always directly proportional to cause.

**Relationships contain feedback loops:** Both negative (damping) and positive (amplifying) feedback are often found in complex systems. The effects of an element's behavior are fed back in such a way that the element itself is altered.

### 3.5. Complex Adaptive Systems

Complex adaptive systems (CAS) are a special category of complex systems to accommodate living beings. As the name suggests, they are capable of changing themselves to adapt to changing environment. The scientific study of CAS has been attempting to find common characteristics and/or formal distinctions among complex systems that might lead to a better understanding of how complexity develops, whether it follows any general scientific laws of nature, and how it might be related to simplicity. Examples of CAS include the stock market (Kogut, Urso et al. 2007; Garas, Argyrakis et al. 2008; Krawiecki 2008; Huang, Zhuang et al. 2009), social insect and ant colonies (Bonabeau 1998), the biosphere and the ecosystem (Levin 1998), the brain and the immune system (Morowitz and Singer 1995), manufacturing systems (Sluga and Butala 2001; Tharumarajah 2003) and any human social group-based
endeavor in a cultural and social system (Wasserman and Faust 1994; Girvan and Newman 2002). The CAS theory also forms the basis of various emergent system models such as the scale free model of the Internet (Faloutsos, Faloutsos et al. 1999) and power grid networks (Albert, Albert et al. 2004), and small world model of social networks (Barabasi and Albert 1999; Amaral, Scala et al. 2000; Newman 2001; Newman 2001; Newman 2001; Barabasi, Jeong et al. 2002) and genetic networks (Bornholdt 2001).

3.6. **Features of Complex Adaptive Systems**

Complex adaptive systems have the following features.

3.6.1. **Agency**

Agents refer to entities (e.g. plants, atoms, or birds) that populate a complex system, and these agents partake in the process of spontaneous change in such a system. However, not all complex systems have agency. In fact, agency, defined as the ability to intervene meaningfully in the course of events (Giddens 1984), is a key characteristic of a CAS. Agents have varying degrees of connectivity with other agents, through which information and resources can flow. Agents have norms, values, beliefs, and assumptions that are shared among their social networks. Agents behave in a manner so as to increase “fitness” of the system that they belong to either locally or globally. Fitness typically refers to the well being of a complex aggregate of both global and local states within the system.
3.6.2. **Self-organization**

Behavior in a CAS is induced not by a single entity but rather by the simultaneous and parallel actions of agents within the system itself. Thus, a CAS is self-organizing and undergoes “a process … , whereby new emergent structures, patterns, and properties arise without being externally imposed on the system” (Zimmerman, Lindberg et al. 1998). This self-organizing behavior can be observed in many systems, whereby simple behavior based on local information can, in the aggregate, lead to complex global behavior.

3.6.3. **Emergence**

Complex adaptive systems follow certain natural laws or rules, but nothing is formally planned in advance so that the behavior of a CAS is emergent. Emergence, is “the arising of new, unexpected structures, patterns, properties, or processes in a self-organizing system … Emergent phenomena seem to have a life of their own with their own rules, laws and possibilities” (Zimmerman, Lindberg et al. 1998).

3.6.4. **Network Connectivity**

A CAS may be described as an aggregate of agents and connections, which suggests that some aspects of the CAS behavior can be deduced by various network theories, such as
social network theory and graph theory (Wasserman and Faust 1994; Choi and Liker 1995). For example, the level of connectivity in the network, in part, determines the complexity of the network. If no connections exist, then agents will behave independently and the aggregate response will be unstructured and random (Dooley and Van de Ven 1999). As connectivity increases, inter-relationships, represented by chains of agents connected together in a contiguous fashion, also increase. The number of inter-relationships is significant because it indicates the potential for the network to engage in global communication from within; it also relates to its potential for chain reactions and effects at a distance.

### 3.6.5. Dimensionality and Feedback

The dimensionality of a CAS is defined as the degrees of freedom that individual agents within the system have to enact their behavior in a somewhat autonomous fashion (Dooley and Van de Ven 1999). The dimensionality of a CAS can be adjusted with feedback. Controls act as a form of negative feedback, effectively reducing dimensionality.

On the other hand, when dimensionality is increased or when a higher degree of autonomy is given to agents to make decisions locally, outcomes are then allowed to emerge in a deviation-amplifying way or through positive feedback.
3.6.6. Dynamism

It has been said that in real-world the only thing constant is change. In fact, the environment of a given CAS is made up of a very large number of other CASs, each of which changes itself and causes others around it to change. One common way changes occur in CASs is through the alteration of the boundaries of the system. These boundaries change as a result of the inclusion or exclusion of particular agents and by addition or elimination of connections among agents, thereby changing the underlying patterns of interactions.

Furthermore, the environment can impose new rules and norms. As a result, the fitness measure may also change because of the changes in the goal state, the basic performance criteria, and/or the way in which the performance criteria are aggregated into a fitness function. As part of the environment, any CAS change would trigger changes in other CASs, and eventually the collective environment changes.

3.6.7. Adaptivity

Typically, significant dynamism exists in a CAS; it necessitates constant adaptation in the CAS. Therefore components of CAS should be able to anticipate and synchronize themselves rapidly with the changing ecosystem and able to adapt easily to forthcoming changes. Furthermore they should be flexible enough to make changes accordingly to both frame of mind and current technologies used.
3.6.8. Rugged Landscape

A dynamic environment surrounds complex adaptive systems that evolve to maximize some measure of goodness or fitness. The potential states that a system in the environment can attain may be represented by a landscape, and the highest point in this landscape may be considered the optimal state for the system.

If the landscape consists of local hills and valleys, meaning that if individual components contribute in different ways, depending on the value/state of other components in a contingent or tightly coupled manner, the optimal state becomes difficult to find, as many local optima exist. Additionally, everything is changing in a dynamic fashion, and so the landscape is dynamic. Therefore, landscapes can be both rugged and changing, forcing the members of the system to both exploit existing knowledge and explore new knowledge (March 1994) in order to overcome the uncertainty imposed on it by the environment and to ensure survivability.

3.6.9. Co-evolution

Complexity theory posits that a CAS both reacts to and creates its environment. A CAS and its environment interact and create dynamic, emergent realities. Because “there is feedback among the systems in terms of competition or cooperation and utilization of the same limited resources” (Zimmerman, Lindberg et al. 1998), the environment forces changes in the entities that reside within it, which in turn induce changes in the other entities and environment itself.
3.6.10. Quasi-equilibrium

Under normal circumstances, a complex system maintains a quasi-equilibrium state, balancing between complete order and incomplete disorder (Goldstein 1994). This balance point, sometimes called the “edge of chaos” (Lewin 1994), allows the system to maintain order while also enabling it to react to qualitative changes in the environment.

In this quasi-equilibrium state, when perturbed by an environmental jolt, the system is normally attracted to its original pattern of behavior. However, sensitivity to environmental changes increases at the interface as the environment pushes the system farther and farther away from the point of quasi-equilibrium. At some point, a state or quantum change occurs, whereby the patterned behavior of the system switches from one attractor to another (Goldstein 1994).

A system that becomes destabilized due to qualitative changes can undergo catastrophic, unpredictable changes that lead it into a different patterned behavior. Such change can be triggered by small (random) exogenous or endogenous fluctuations when the system is far from equilibrium.


3.6.11. Non-linearity

Behavior in a complex system stems from the complex interaction of many loosely coupled variables therefore the system behaves in a non-linear fashion. In other words, since systems behave in non-linear ways, these behaviors naturally exhibit a non-linear response to changes. Therefore, in a non-linear system, large changes in input may lead to small changes in outcome, and small changes in input may lead to large changes in outcome (Guastello 1995). Thus, in practice, a complex system can be hypersensitive to small changes in its environment.

A simple adaptive response, which usually leads to a simple corrective action, can lead to a catastrophic outcome (the so-called “butterfly effect”). Even if initial conditions and generative mechanisms are exactly specified (which they cannot be), prediction of the future often becomes fruitless as specification errors grow exponentially as one progresses into the future (Peitgen, Jurgens et al. 1992). Therefore the behavior of a complex system cannot be written down in closed form; it is not amenable to prediction via the formulation of a parametric model, such as a statistical forecasting model.

3.6.12. Non-random Future and Path Dependence

The inability to determine the future behavior of a complex system in an exact manner, however, does not imply that the future is random. This observation may appear paradoxical at first glance, but complex systems, in fact, exhibit patterns of behavior that can be considered archetypal or prototypical. It is true that small changes may lead to
drastically different future paths; however, the same characteristic pattern of behavior
emerges despite the change, meaning that they are path dependent (Mahoney 2000;
Garrouste, Ioannides et al. 2001). One finds that systems will tend to be involved in
certain prototypical ways and, thus, our predictive capacity, although limited to the exact
prediction at a future point in time, can benefit from the knowledge of these patterns.

3.6.13. Difficulty to Determine Boundaries

It is difficult to determine boundaries of a CAS. Depending on the scale of analysis, an
agent may represent an individual, a project team, a division, or an entire organization.
Different sizes of boundaries can be studied for the same CAS, according to the choice of
agents, norms, rules and level of connectivity with other agents.


A CAS is highly structured. The components of a CAS are self-contained within larger
components. In other terms the components of CAS are complex systems as well.
Chapter 4

Network Theory

In the context of network theory, a complex network is a network (graph) with non-trivial topological features—features that do not occur in simple networks such as lattices or random graphs (Barabási 2002). The study of complex networks (large scale networks) first has been suggested to explain different features exhibited by natural complex systems; however later it is inspired largely by the empirical study of real-world networks such as computer networks and social networks.

Traditionally, network theory has focused on static networks, i.e., networks that do not change their structure over time or as a result of agent behavior. Recently, much progress has been made in understanding the growth and change of real-world networks (Barabási 2002). In particular “small world” or scale-free networks have been discovered in a wide range of settings. Dynamic network analysis is a growing field that incorporates the mechanisms of network growth and change based on agent interaction processes (Brieger, Carley et al. 2003). Understanding the agent rules that govern how networks are structured and grow, how quickly information is communicated through networks, and the kinds of relationships that networks embody are important.
Two well-known and much studied classes of complex networks are scale-free networks (Albert, Jeong et al. 1999; Faloutsos, Faloutsos et al. 1999; Adamic, Lukose et al. 2001; Albert, Albert et al. 2004) and small-world networks (Pastor-Satorras and Vespignani 2001; Pastor-Satorras and Vespignani 2001; Montoya and Sol 2002; Newman, Jensen et al. 2002; Pastor-Satorras and Vespignani 2002; Krawiecki 2008) which have many local interactions and a smaller number of inter-area connections are often employed. Both are characterized by specific structural features – a heavy tail in the degree distribution (power-law degree distributions) for the former and short path lengths and high clustering for the latter. However, as the study of complex networks has continued to grow in importance and popularity, many other features of network structure have attracted attention as well. Such features include assortativity or disassortativity among vertices (Newman 2002), community structure (Amaral, Scala et al. 2000; Girvan and Newman 2002; Clauset, Newman et al. 2004; Newman and Girvan 2004), and hierarchical structure (Clauset, Moore et al. 2008). In the case of directed networks (Karrer and Newman 2009) these features also include reciprocity, triad significance profile and other features. In contrast, many of the mathematical models of networks that have been studied in the past, such as lattices and random graphs, do not show these features.

### 4.1. Modeling of Complex Networks

#### 4.1.1. Random Graphs

In mathematics, a random graph is a graph that is generated by some random process. Erdos and Reiny (ER) random graphs were used to model complex networks for a long
time. The model is intuitive and analytically tractable; moreover the average path length of real networks is close to the path length of an ER random graph of the same size. However, recent studies showed that large scale networks have significantly different properties compared to ER random graphs. It has been found that the average clustering coefficient of real networks is significantly larger than the average clustering coefficient of ER random graphs with the same number of nodes and edges, indicating a far more ordered structure in real networks. Moreover, the degree distribution of many large scale networks are found to follow a power law distribution, $p(k) \sim k^{-\gamma}$.

### 4.1.2. Small World Networks

A network is called a small-world network by analogy with the small-world phenomenon (popularly known as six degrees of separation). The small world hypothesis, which was first described by the Hungarian writer Frigyes Karinthy in 1929, and tested experimentally by Stanley Milgram (1967), is the idea that two arbitrary people are connected by only six degrees of separation, i.e. the diameter of the corresponding graph of social connections is not much larger than six.

In 1998, Duncan Watts and Steven Strogatz published the first small-world network model, which through a single parameter smoothly interpolates between a random graph to a lattice. Their model demonstrated that with the addition of only a small number of long-range links, a regular graph, in which the diameter is proportional to the size of the network, can be transformed into a "small world" in which the average number of edges
between any two vertices is very small (mathematically, it should grow as the logarithm of the size of the network), while the clustering coefficient stays large. It is known that a wide variety of abstract graphs exhibit the small-world property, e.g., random graphs and scale-free networks. Further, real world networks such as the World Wide Web and the metabolic network also exhibit this property.

Small world refers to the co-occurrence of a small diameter and a high clustering coefficient. The clustering coefficient is a metric that represents the density of triangles in the network. For instance, sparse random graphs have a vanishingly small clustering coefficient while real world networks often have a coefficient significantly larger. Scientists point to this difference as suggesting that edges are correlated in real world networks.

4.1.3. Scale Free Networks

A network is named scale-free if its degree distribution, i.e., the probability that a node selected uniformly at random has a certain number of links (degree), follows a particular mathematical function called a power law. The power law implies that the degree distribution of these networks has no characteristic scale. In contrast, network with a single well-defined scale are somewhat similar to a lattice in that every node has (roughly) the same degree. Examples of networks with a single scale include the Erdős-Rényi random graph and hypercubes. In a network with a scale-free degree distribution, some vertices have a degree that is orders of magnitude larger than the average – these
vertices are often called "hubs", although this is a bit misleading as there is no inherent threshold above which a node can be viewed as a hub.

Interest in scale-free networks began in the late 1990s with the apparent discovery of a power-law degree distribution in many real-world networks such as the World Wide Web, the network of Autonomous systems (ASs), some network of Internet routers, protein interaction networks, email networks, etc. Although many of these distributions are not unambiguously power laws, their breadth, both in degree and in domain, shows that networks exhibiting such a distribution are clearly very different from what one would expect if edges existed independently and at random (a Poisson distribution). Indeed, there are many different ways to build a network with a power-law degree distribution. The Yule process (Yule 1925) is a canonical generative process for power laws recently renamed as preferential attachment by Barabási and Albert (1999) for power-law degree distributions.

Networks with a power-law degree distribution can be highly resistant to the random deletion of vertices, i.e., the vast majority of vertices remain connected together in a giant component. Such networks can also be quite sensitive to targeted attacks aimed at fracturing the network quickly. When the graph is uniformly random except for the degree distribution, these critical vertices are the ones with the highest degree, and have thus been implicated in the spread of disease (natural and artificial) in social and communication networks, and in the spread of fads (both of which are modeled by a percolation or branching process).
4.2. Statistical Properties of Complex Networks

In this dissertation, we investigate the computer manufacturing enterprise network structure by computing network properties. Some of the properties that are analyzed in our study are described in this section.

4.2.1. Network Demographics

Network demographics of a network include the number of nodes, the number of edges, the number of weak component clusters, strong component clusters, self-loops, parallel edges, symmetry of graph (whether the network appears to be directed or undirected), and the attributes present on both nodes and edges.

4.2.2. Network Density

Measuring the density of a network gives us a ready index of the degree of dyadic connection in a population. For binary data, density is simply the ratio of the number of adjacencies that are present divided by the number of pairs – what proportion of all possible dyadic connections are actually present. If we have measured the ties among actors with values (strengths, closeness, probabilities, etc.), density is usually defined as the sum of the values of all ties divided by the number of possible ties. That is, with valued data, density is usually defined as the average strength of ties across all possible
(not all actual) ties. Where the data are symmetric or un-directed, density is calculated relative to the number of unique pairs \( (n^*n-1)/2 \); where the data are directed, density is calculated across the total number of pairs.

### 4.2.3. Average Degree

The average degree is a measure that is based on the degree of nodes in the network. It shows how many neighbors (adjacent nodes) a node in the network has on average. This measure can only be calculated when the network has at least one edge connecting the nodes. The average degree of a complex adaptive supply network reflects the average volume of information flow and sharing that may have influenced the system at hand.

### 4.2.4. Scale Free Characteristic

As stated in the modeling of complex networks, a network is named scale-free if its degree distribution, i.e., the probability that a node selected uniformly at random has a certain number of links (degree), follows a particular mathematical function called a power law. It is reported that most of the real-world networks including the World Wide Web (Albert, Jeong et al. 1999; Huberman and Adamic 1999; Kumar, Raghavan et al. 2000), the Internet (Faloutsos, Faloutsos et al. 1999), metabolic networks (Jeong, Tombor et al. 2000), phone call networks (Abello, Pardalos et al. 1999; Aiello, Chung et al. 2000), scientific collaboration networks (Newman 2001; Newman 2001; Newman 2001), and movie actor collaboration networks (Faloutsos, Faloutsos et al. 1999; Amaral, Scala et al. 2000; Barabasi, Jeong et al. 2002) follow a power-law degree distribution, \( p(x) \sim x^{-\gamma} \). The
presence of the power-law distribution indicates that the topology of the network is highly heterogeneous, with a large fraction of small-degree nodes and a small fraction of large-degree nodes.

4.2.5. Average Shortest Path Length

The average shortest path length of a connected network is the average of the shortest paths that exist between all possible pairs of nodes in a network. It is given by

$$l \equiv \langle d(u,v) \rangle = \frac{1}{N(N-1)} \sum_{u \in V} \sum_{v \neq u \in V} d(u,v)$$

(4.2.5.1)

where, $N$ is the number of nodes and $V$ is the set of nodes in the network; and $d(u,v)$ is the shortest path between nodes $u$ and $v$ such that $u \neq v$ in the network.

4.2.6. Clustering Coefficient

Watts and Strogatz (Watts and Strogatz 1998) introduced the important concept of clustering in social networks. The clustering coefficient characterizes the local transitivity and order in the neighborhood of a node. The clustering coefficient of node $i$ is the ratio of the number of edges $E_i$ that actually exist among the one-link-away neighborhood nodes to the maximum number of possible edges between these neighbor nodes. For a directed graph, the edge, $e_{ij}$, connecting node $i$ with node $j$ is different from the edge, $e_{ji}$,
connecting node \( j \) with node \( i \). Therefore if node \( i \) has \( k_i \) one-link-away neighbors there could exist at most \( k_i(k_i-1) \) links among the neighbor nodes. Thus, the clustering coefficient of node \( i \) in a directed graph is given by

\[
C_i = \frac{E_i}{k_i(k_i-1)}
\]  (4.2.6.1)

For an undirected graph, the edge \( e_{ij} \) is identical to the edge \( e_{ji} \). Therefore at most \( k_i(k_i-1)/2 \) edges could exist among the nodes within the neighborhood of node \( i \). Thus, the clustering coefficient of node \( i \) in an undirected graph is given by

\[
C_i = \frac{2E_i}{k_i(k_i-1)}
\]  (4.2.6.2)

The clustering coefficient of the entire network \( C \) is the average of \( C_i \)'s over all the nodes in the network, i.e.,

\[
C = \frac{1}{n} \sum_i C_i
\]  (4.2.6.3)
4.2.7. Small World Network Characteristics

Observing short path length and high clustering in a complex network indicates small world characteristics. The existence of the small-world effect was first demonstrated by the famous experiment conducted by Stanley Milgram in the 1960s (Milgram 1967). It led to the popular concept of six degrees of separation. Milgram found that there is an acquaintance path of an average length of 6 in the social network of people in the US.

4.2.8. Percolation Theory

One of the most interesting properties of graph theory is percolation transition. According to this theory, there exists a critical probability $p_c$ below which the network is a collection of isolated clusters and above which the network forms a giant component quite rapidly (Albert and Barabasi 2002). By using analogy of percolation transition, connectivity of a complex network can be analyzed via its largest connected components.

4.2.9. Diameter

The diameter of a network is the longest shortest path between two vertices. So one considers every pair of vertices, finds the shortest path between each of the pairs, and then the longest of these shortest paths determines the diameter. The diameter of the network indicates the greatest path length/distance a node will ever have to hop to reach another node in the network. For the computer manufacturing enterprise network,
diameter means that the length of the chain of information flow connecting any two enterprise is less than or equal to a constant.

4.2.10. Community Structure

There are many definitions of a “community” in the literature (Wasserman and Faust 1994; Flake, Lawrence et al. 2000; Newman and Girvan 2004; Radicchi, Castellano et al. 2004; Palla, Derenyi et al. 2005), but not a unique one due to the ambiguity of how dense a group should be to call it a community. A simple definition given by Flake, et al. (Flake, Lawrence et al. 2000) and Radicchi, et al. (Radicchi, Castellano et al. 2004) considers a subgraph as a community if each node in the subgraph has more connections within the community than with the rest of the graph.

Newman and Girvan (Newman and Girvan 2004) have proposed another measure which is calculated as the fraction of the total network links that are within the community minus the analogous fraction in a randomized counterpart of the network: the higher the difference, the stronger the community structure. One of the fastest community structure algorithms is "Fast Modularity Community Structure Inference Algorithm” developed by Clauset et al. (Clauset, Shalizi et al. 2009) to identify communities in a network. This algorithm runs in $O(nd \log n)$ time for sparse graphs where $n$ is the number of nodes in the network and $d$ is the depth of the dendrogram describing the network’s community structure.
Chapter 5

Defining System Structure and Characteristics

5.1. Supply Chain

A supply chain is the flow and management of resources across the enterprise for the purpose of maintaining the business operations profitably (Sehgal 2009). A supply chain is a system of organizations, people, technology, activities, information and resources involved in moving a product or service from supplier to customer. Supply chain activities transform natural resources, raw materials and components into a finished product that is delivered to the end customer.

In a supply chain, materials may refer to raw materials, work-in-process, or finished products. Generally a material has purchasing cost, storing cost and distribution cost. While overstocking of materials has storing costs, out of stock causes loss of sales and customers. People; managers, labors and customers; are all key components of a supply chain. Management of the flow of information is equally important in a supply chain. Mismanaged, inaccurate or asynchronous information results in customer dissatisfaction. Responding to customer demand within the supply chain is not possible without proper management of the information flow.
Flow and management of resources have significant importance in supply chain management. Some of the resources flow through the supply chain, such as merchandise. It flows from suppliers’ warehouses to the retailers’ warehouses, then to stores and customers. Other resources, such as workers in the distribution center, take part in the flow process of the merchandise. The main objective of a well managed supply chain is to plan and execute the flow and management of the resources for the purpose of maintaining the business operations profitability. However, resources are generally scarce and cost money. Therefore, if an enterprise has a supply chain which is more robust and profitable than its competitors, it will be affected less from environmental disturbances. If not, enterprise reconfiguration in terms of utilization of triple-A measures might be necessary.

In general, a supply chain has two ends: the demand end and the supply end. The demand end of the supply chain models elements of the supply chain where the demand originates. The demand end is also called as downstream and contains stores and customers. The supply end of the supply chain, also called upstream, represents the sources of supply, such as warehouses or factories. More clearly, upstream represents the supply chain elements that provide supplies to address the demand generated at the downstream. Thinking of the supply chain in terms of demand end and supply end helps in understanding the problem of balancing supplies against demand.

Demand flows from downstream nodes to the upstream nodes in a supply network. Similarly, the supplies flow from upstream nodes to downstream nodes. Between the supply and demand ends of a supply chain, there might be warehouses, cross-docking facilities,
transshipment points, processing facilities, assembly plants, etc. All those elements – downstream elements, upstream elements and the elements between these two – are the nodes of a supply chain.

The relationships among those elements are called flows. The network of nodes and flows in the supply chain model creates the supply chain network. Supply chain network processes help in evaluating the manufacturing, distribution, supply, and sales networks. As enterprises grow, supply chain efficiencies are impacted due to the changes in demand and supply patterns. The network planning functions provide methods for evaluating the current networks and determining the optimal design for future networks using the forecasted or planned demand and supply projections.

Network design and optimization for supply chains allow corporations to plan their growth to be aligned with their business strategy, growth targets, and projected changes in demand for their products. Implementing any changes in the existing network is capital and time intensive. Therefore network evolution and optimization remains a decision support process that should be part of a business planning exercise.

We presented an overview of the organization of supply chain functions. We will cover the decision support process that help in longer term planning and forecasting; demand planning and supply planning. Then we will cover the processes that help run the day-to-day operations in a supply chain. Supply chain management, transportation, warehousing, and reverse logistics execution fall into this section.
5.1.1. Demand Planning in Supply Chains

Demand planning is the process of forecasting the demand for products in a way that the demand can be fulfilled through existing inventory, manufacturing and new purchases. Demand planning is one of the most important supply chain processes in the sense that it drives almost all other processes directly or indirectly toward fulfilling the demand.

The projected demand determines what a retailer should buy or a manufacturer should build. This in turn drives the factory capacity and resource utilization, raw material demand, and orders on vendors for such raw material. The demand is projected at the downstream and gets propagated through the supply network until it is satisfied with the supply from a supplying node. If the demand can be satisfied from the available inventory, then the demand propagation stops at that point. If there are not enough supplies in the inventory to meet all demand, then a demand request is sent to the next upstream node in the supply chain. Demand planning consists of various subprocesses:

**Demand Forecasting:** Demand forecasting provides the projected future demand usually based on the historical data for the specified product–location combination.

**Allocation Planning:** Allocation planning is part of the demand management process that allows for managing seasonal merchandise in an effective manner.

**Replenishment Planning:** Replenishment planning takes the unconstrained demand forecasts as input and generates the replenishment plan, taking the available inventory into consideration.
Demand planning answers the following questions:

- What is the unconstrained projected demand for a product?
- What is needed to fulfill a demand? How much inventory should be kept at various points on the supply network to ensure that the demand at all consuming nodes can be fulfilled?

5.1.2. Supply Planning in Supply Chains

The supply planning processes complement the demand planning functions. Supply plan is created after the net demand is determined. The subprocesses of supply planning are described below.

**Inventory Planning:** The inventory planning process establishes the optimal inventory levels that must be maintained to meet expected fulfillment service levels. Any two nodes in a supply chain can be viewed as having a supplier–consumer relationship as the material flows from one node to another. When viewed as such, the node that acts as a consumer is placing demand on the node that acts as a supplier. This demand must be fulfilled by the supplying node at a user specified fulfillment service level. To guarantee such service levels, the supplying node must maintain an optimal level of inventory. Artificially high service levels will push these inventory levels too high, and result in unusually high inventory costs and low inventory turns. The relationship between the amount of inventory required and service level is exponential, and therefore every little improvement in service levels will push the inventory levels higher.
**Replenishment Planning:** Replenishment planning considers the on-hand inventories, expected receipts / inbound inventory, and expected shipments / outbound inventory for the planning period under consideration. Here net demand is simply the forecast demand minus on-hand minus expected receipts plus expected shipments.

**Production Planning:** Production planning process helps align the manufacturing resources with the demand in the supply chain that has been identified to be fulfilled through factory orders. The production planning process reviews the demand of the finished goods to be fulfilled and establishes the raw material and manufacturing capacity requirements.

**Logistics Capacity Planning:** Logistics capacity planning is focused on longer-term transportation and warehousing capacity planning. It consists of analyzing the current and projected needs for transportation and warehousing based on the enterprise’s business plans for the future. The transportation planning process ensures that enough capacity is available under contract with the carriers on the projected routes to address the changing demands of shipping between suppliers, factories, warehouses, and stores. The warehousing capacity planning ensures that there is enough warehousing capacity to stock the projected inventory levels to serve the planned future business volumes.

### 5.1.3. Supply Management in Supply Chains

The supply management functions of a supply chain have a large scope from sourcing to purchasing, manufacturing, replenishment, and vendor performance management. On a
broader level, there are two aspects of supply management functions. The first deals with supply management functions such as determining the best sources for merchandise, the procurement of the merchandise, and global trade. For example, the strategic sourcing process consists of finding out the best sources of supply, determining the feasibility of a strong and lasting relationship with the vendor, and managing this relationship over time to mutual benefit and advantage.

The second part of the supply management functions deals with the suppliers themselves and managing the relationship with them. For example, retailers deal with many suppliers to replenish their warehouses and stores. Manufacturers need predictable supplier bases to guarantee raw material supplies without interruption to ensure efficient utilization of their resources and ability to service their customers reliably. Therefore, establishing consistent and good supplier-partner relationships and good management of those relationships provide opportunities to optimize the total cost, source of supplies and delivery schedules.

5.1.4. Management of Other Resources in Supply Chains

*Transportation Management:* Transportation planning and execution processes help in optimally moving the merchandise from one location to another. Any business process that needs merchandise movements from one position to another can be optimized by using the transportation planning functions.
**Warehouse Management:** Warehousing is an equally important component of the distribution equation along with the transportation. Warehouses provide the locations where inventory is stocked primarily to absorb demand fluctuations and to provide smoother operations of the supply chain. Warehouse management process address the functions required to efficiently manage operations such as receiving inventory, stocking and tracking inventory, and shipping it to stores when required.

**Reverse Logistics Management:** Reverse logistics management covers all the business functions that allow a retailer to process the merchandise returns generated at the stores, web sites, or warehouses. Reverse logistics management consists of managing the flow of merchandise from stores and customers back to supplier. The complete returns transaction can contain a few shipment legs, warehousing, packing, handling, and other warehouse activities. Due to the complexity of managing the reverse flow, many enterprises simply subcontract the reverse logistics to a third-party logistics provider.

**5.2. Triple-A Measures**

Today’s business is witnessing unprecedented changes. New products, new processes, new technologies, new markets, and even new competitors are appearing and disappearing within short periods of time. Historically, mass production has slowly led to lean production. Then lean production evolved into agile production. In the 1950s enterprises focused on productivity improvement and in the 60s and the 70s they concentrated on quality enhancement. While in the 80s enterprises worked hard to achieve flexibility, in the 90s they
are occupied with agility and environment. Today only being agile is not enough; enterprises need two additional measures: adaptivity and alignment. Let’s define and understand these three measures in more detail.

5.2.1. Agility

First measure, agility, is the capability of rapidly and efficiently adapting to changes. It requires rapid decision making and execution and being nimble to new market and technological trends. Agility is the ability of producers of goods and services to thrive in rapidly changing, fragmented markets. Agility is a comprehensive response to the challenges posed by a business environment dominated by changes and uncertainties.

An agile company should be capable of operating profitably in a competitive environment of continually and unpredictably changing customer opportunities (Goldman, Nagel et al. 1995). Agility is one of the three measures that an enterprise should possess to survive environmental disturbances in today’s business markets.

Agility can be achieved using different techniques according to the type of business environment in which they thrive. Agility can be achieved by staying close to the customer to provide value-based solutions in an enterprise that is driven by mass customization ideas. Examples of this type of enterprise systems include customized jeans by Levi-Strauss and customized trucks by Mack Trucks.
Agility can be also achieved by awareness of predictable changes in technology, sales, markets, and so on. Evolution of laptop computers, for example, is driven by predictable changes in the technology; laptops will become thinner, weigh less, pack more power, and incorporate newer features. Sales surge of air conditioners and electric fans in summer and sales increase of candies during Christmas are good examples of predictable changes in demand.

Companies, which are not agile enough, are not able to anticipate fundamental marketplace shifts. Nearly 90% of executives surveyed by the Economist Intelligence Unit believe that organizational agility is critical for business success (Economist 2009).

5.2.2. Alignment

Second measure we study in this dissertation is alignment of enterprises. Alignment refers to the synchronization of shared goals and objectives of a set of companies in an ecosystem. Today, it is not enough to set clear, realistic goals and share those with the market. Complex market structure requires developing concrete strategies and effective communication tactics to achieve alignment goals and bring value to the enterprise. Alignment also requires a step-by-step mapping of an enterprise’s current, actual supply chain and a comparison to goals that have been set out for this enterprise’s optimal supply chain and also the standards set by its competitors.
Market share, customer satisfaction, taking care of employees, innovation and innumerable other virtues are essential to secure the market position of an enterprise. They are desirable, but only as the means for reaching the objective of creating value that are ultimately measurable only as a favorable financial outcome for a commercial enterprise. However, alignment of companies has many benefits such as increased sales, higher profits, lower costs, lower turnovers, satisfied employees and better customer service. Failure of alignment may cause decrease in sales and profits, missed delivery dates, unmatched priorities and therefore chaos in the system.

*Alignment in Computer Manufacturing Enterprise Ecosystem:* In addition to the inherent complexity of computer manufacturing enterprise ecosystem, it is also hard to coordinate the actions of enterprises across their organizational boundaries so that they perform in a coherent manner. But why do we need aligned enterprises that perform in a coherent manner? Aligning computer manufacturing enterprises to corporate goals has emerged as one of the number one concerns over the last decade with the effect of fast pace technology and shortening average product life.

To line up computer manufacturing enterprises with market expectations requires the realization of the common goals. Common goals of the enterprises should be reflected to customers in a coherent manner. The important measures to be aligned are customer satisfaction, quality and workflow simplification, which sometimes do not contribute to net cash flow. Therefore, alignment efforts should include a positive relationship between these measures and the accepted financial measures of performance.
5.2.3. Adaptivity

The third measure is adaptation, which in biology is defined as the evolutionary process, which takes place over many generations, whereby a population becomes better suited to its habitat. An enterprise ecosystem is a complex system, where significant dynamism exists, and necessitates constant adaptation. In this ecosystem, enterprises are concerned themselves with adaptability, the ability to change when an unexpected change occurs or an unforeseen opportunity appears because emergence of new and non-traditional competitors and increasing customer demand unsettle the current position of companies in the market. For example a sudden outbreak of war may lead to many problems in operating a business. The key factors of being sustainable and competitive are being able to respond timely and being flexible enough to adapt new technologies and ideas. The enterprises which neglect the change become irresponsible and are not able to use the opportunities that come into play every day in business. Enterprises should embrace change to keep moving forward and keep the company’s competitive edge.

In the enterprise ecosystem, building adaptive capabilities is more challenging than creating agile capabilities for an enterprise.
Chapter 6

Formulation of an Enterprise System as a Complex Adaptive System

In this chapter we use the concepts of complex adaptive systems to formulate the enterprise ecosystem structure and explain the dynamics of rapidly changing business ecosystem that affect enterprise success and survival. We explore that nature once again provides the answer to what it seems complex and difficult to conceive. We list the characteristics of the complex adaptive systems and how they apply to CMEE. The study of enterprise systems from a complex adaptive perspective has a significant contribution since contiguity of a product or system in today’s market is as important as its innovation and putting it into practice.

In general, managers have come to realize the importance of system structure, information flow and system optimization (Cavinato 1992; Ellram 1993; Lee and Billington 1993). Consequently, many enterprises have spent increasing amounts of time, money, and effort in an attempt to predict and control their future. However, enterprises’ efforts to manage enterprise systems, including CMEE, have often led to frustration and helplessness. Managers have struggled with the dynamic and complex nature of enterprise ecosystem and the inevitable lack of prediction and control. For instance, rapid changes that occur at different parts of the ecosystem are often outside the purview and control of a single enterprise.
CMEE is a diverse, heterogeneous, and hierarchical ecosystem that evolves in a sensitive environment to initial conditions or to small perturbations. A typical CMEE with a tree-like or hierarchical structure lacks the basic survivability, flexibility, scalability and adaptivity properties. This heterogeneity of CMEE is based on collaboration and information links that contains diverse interaction variables. Therefore we need a model of CMEE that more accurately reflects its underlying complexity and dynamism. We propose it is not enough to recognize a CMEE as simply a system — a CMEE should be treated as a complex adaptive system (CAS). In a business context, a CAS is not just a question of selection and evolution, but of complex interactions among the people and structures of a business and its marketplace. In this context organizations mutated into organisms, markets to ecologies, and mechanical metaphors to biological analogues.

The concept of complexity allows us to understand how CMEE co-evolves, and it can help us to identify the patterns that arise in its evolution. Because complexity theory augments traditional systems theory (i.e., it emphasizes time and change), it can be used to help us to recognize change in the enterprises within the CMEE, change within the market, change within the inter-relationships among enterprises, and change in the environment. This complex web of changes, coupled with the adaptive capability of organizations within the CMEE to respond to such changes, lead to self-organization within the network. Order and control in the network are emergent, as opposed to predetermined. Control is generated through simple (but non-linear) behavioral rules that operate based on local information.
6.1. Agency

In the CMEE, individual enterprises in computer and peripheral manufacturers, semiconductor and other electronic component manufacturers, navigational, measuring, electro-medical, and control instruments manufacturers, magnetic and optical media manufacturers and software enterprises such as Microsoft and Macintosh are agents that constitute the nodes in the system. Their actions affect the course of events that occur in the CMEE — they select and de-select suppliers, choose to move into new markets, and adopt integrated processes of product development involving suppliers and customers. Further, the nature of relationships among these enterprises with their buyers and suppliers varies (i.e. long-term versus short-term). These agents or enterprises attempt to increase “fitness” by attending to a few simple dimensions such as delivery, cost, quality, and flexibility.

In a business system, fitness is a measure of the health of an organization. A company’s fitness is a function of how well it is able to respond to and anticipate the changes in its business environment. One should keep in mind that niches are created by the interactions of multiple agents; the greater the number of agents and interactions, the greater and richer the diversity of the environment and agents. Thereby with the effect of greater and richer population diversity, the likelihood that the interactions will generate individuals with better fitness values increases.

In an ecological context, diversity refers to the number of different species that inhabit an ecosystem. Diversity is a measure of a system’s variety: the greater the variety, the greater
the diversity. In a business system, diversity can be regarded as a form of economic and social wealth because the greater the diversity, the greater the economic activity, therefore, the greater the opportunity for new forms of economic growth through the emergence of new niches or markets. Consequently, the greater the diversity of a CAS, the more fit it is. For instance, a computer manufacturing enterprise tries to increase fitness by ensuring the acceptable cost, delivery, and quality purchased, by building a good network of business partners. As the enterprises share their common norms and procedures, transaction costs reduce and communication efficiency increases.

6.2. Self-organization

Self-organization is the process where a structure or a pattern appears in a system without a central authority or external element precipitating it through planning. This global coherent pattern appears from the local interaction of the elements that make up the system, thus the organization is achieved in a way that is parallel (the entire elements act at the same time) and distributed (none of the elements is a central coordinator).

Self-organization, as opposed to natural or social selection, is a dynamic change within the organization where system changes are made by recalculating, re-inventing and modifying organization structure in order to adapt, survive, grow, and develop. Self-organization is the result of re-invention and creative adaptation due to the introduction of or being in a constant state of perturbed equilibrium. One example of an organization which exists in a constant state of perturbation is that of the learning organization, which is "one that allows self-
organization, rather than attempting to control the bifurcation through planned change” (Dooley and Van de Ven 1999).

From the view of each individual computer manufacturing enterprise, a CMEE is a self-organizing system. Although the totality may be unknown, individual computer manufacturing enterprises partake in the grand establishment of the CMEE by engaging in their localized decision-making (i.e., doing their best to select capable suppliers and ensure on-time delivery of products to their buyers).

In a system, biological or business, chaos eventually gives way to self-organization. The question is how to control the duration, intensity, and shape of its outcome. It seems that punctuating equilibrium and instilling disorder in an enterprise system is a risky business. Throwing an enterprise off balance could possibly send it in a downward spiral towards dissemination by ultimately compromising the structural integrity (i.e., identity) of the system to the point of no return.

The only way to reap the benefits of chaos theory while maintaining a sense of security is to adjust the organization towards a state of existence which lies “on the edge of chaos”. By existing on the edge of chaos, organizations are forced to find new, creative ways to compete and stay ahead. Being "off-balance" lends itself to regrouping and re-evaluating the system’s present state in order to make needed adjustments and regain control and equilibrium.
Good examples of such learning organizations are found throughout the field of technology as well as the airline industry, namely organizations such as Southwest Airlines, which used re-invention not just for survival, but also to prosper in an otherwise dismal market. In contrast, there are organizations which, due to extended periods of equilibrium, find themselves struggling for survival. Telephone companies, for instance, were once solid and static entities dominated the communication market. While the rest of the world was developing new communication technology, telephone companies did not creatively grow at the same rate. The result is an organization that is battling to stay alive unless they embrace the element of chaos due to crisis, and allow creative adaptability to function freely so that self-organization and re-invention can occur.

### 6.3. Emergence

The behavior of the CMEE is said to “emerge” from the actions of its enterprises. In general system and also individual enterprises emerge to gain a deterministic operating environment. This emergence of highly structured collective behavior over time from the interaction of the simple entities leads to fulfillment of orders. However, we also observe undesirable emergent phenomena when factors, such as sudden technology advancements, changing social trends, and conflicting objectives between enterprises come into the picture in a CMEE. Those emergent phenomena may cause demand decline or amplification and therefore inventory swings. Further it may increase competition that arises in the form of sharing and contention of resources. As a conclusion global control over agents is an exception rather than a rule;
more likely is a localized cooperation out of which a global order emerges, which is itself unpredictable.

6.4. Network Connectivity

CMEE classifies into denser cluster of smaller industries, each of this clusters has dense network connections across the clusters. For example, some of those network connections lie across the globe. Many products are being designed in one country, manufactured in another, and assembled in a third. In the computer manufacturing enterprise ecosystem, the United States industry tends to focus on high-end products, such as computers and microchips. Even so, many components of final products manufactured in the United States are produced elsewhere and shipped to a U.S. plant for final assembly.

Despite dense network connections across the clusters, the different enterprises in a cluster typically operate autonomously with different objectives and subject to different set of constraints. However, when environmental constraints get harder to satisfy such as improving due date performance, increasing quality or reducing costs, they become highly inter-dependent. It is the flow of material, resources, information and finances that provides the binding force.

The welfare of any enterprise in the system directly depends on the performance of the others and their willingness and ability to coordinate. This leads to correlations between entities
over long length and timescales. For example, most electronic products contain many intermediate components that are purchased from other manufacturers.

Companies producing intermediate components and finished goods often choose to locate near each other so that companies can receive new products more quickly and lower their inventory costs. It also facilitates joint research and development projects that benefit both companies. As a result, several regions of the country have become centers of the electronic products industry. The most prominent of these centers is Silicon Valley, a concentration of integrated circuit, software, and computer firms in California's Santa Clara Valley, near San Jose.

However, there are several other centers of the industry throughout the country (Bureau of Labor Statistics and U.S. Department of Labor 2011). In that sense CMEE are complex systems, using “complex” in the technical sense to mean that the behavior of the system as a whole cannot be determined by partitioning it and understanding the behavior of each individual enterprise separately. Therefore studies in CMEE should focus on the entire network rather than interactions of isolated enterprises in the network (i.e. single buyer and single supplier). In this regard, we can consider a CMEE as the population of all involved enterprises and the actual computer manufacturing enterprises as depicting the connectivity between agents (enterprises).

CMEE possess a massively complex, interconnected structure. Realistically, a critical level of connectivity exists such that there are inter-connections among enterprises that otherwise
would assume themselves to be decoupled. Within a CMEE, enterprises that are aware of activities across the different parts of the network will be more effective at managing materials flow and technological developments than enterprises that are aware of activities of only their immediate network.

For example, many of the companies in the navigational, measuring, electro-medical, and control instruments manufacturing segment work as government contractors, producing equipment for military purposes. In some cases, this technology has been adapted to computer and peripheral manufacturing segment for consumer use. For example, GPS technology was originally designed for use by the U.S. Navy, but has been developed into a navigation system that individuals can use in their cars, phones or computers.

6.5. Dimensionality and Feedback

An important element in managing CMEE is to control the ripple effect of response time so that the negative effects of environmental disturbances are minimized and agility and robustness of the enterprise system are maximized. Control of ripple effect requires coordination between entities in performing their tasks. With the increase in the number of participants in the CMEE, the problem of coordination gets harder. This is the reason why CMEE is complex; it consists of a large number of independent interacting enterprises via their supply-demand, collaboration and information flow links. These links contain both positive and negative information feedback. Slow and delayed feedback weakens CMEE and leaves the system unguarded against unpredicted rapid changes. Yet the need for control in a
CMEE is real because a system with its high dimensionality tends to be difficult to control. For example, no matter how hard IBM tries to control its DRAM chip suppliers by imposing co-location and price requirements, it is impossible to control the global economic dynamics in the chip manufacturing industry.

Control is a form of negative feedback that effectively reduces dimensionality. Successful implementation of control oriented norms and processes (e.g. ERP, JIT II) leads to higher efficiencies, but it may also lead to negative consequences such as less than expected performance improvements and reduction in innovative activities by the suppliers. When institutional controls are insufficient for controlling outcomes, another approach is to seek places where the transfer of cause and effect is relatively flat or to seek system designs that are robust in the Taguchi sense (Choi, Dooley et al. 2001).

Conversely, positive feedback, which increases dimensionality, produces a more creative and adaptive response from the system. For example, Nishiguchi (Nishiguchi 1994) documented that when suppliers are given autonomy to think creatively, they can come up with innovative ideas to improve current product configurations.

Globalization also plays a major role in dimensionality of CMEE, often making it difficult to distinguish between American and foreign companies. Many U.S. companies are opening plants and development centers overseas and overseas companies are doing the same in the United States.
6.6. Dynamic Process

An important characteristic of CMEE is that it is a result of dynamic processes. Each enterprise within the CMEE is constantly “in motion”: doing industrial/business type services, operating and transforming. CMEE emerges from this constant change. CMEE only exists while its members are transforming, adapting and synchronizing. So that CMEE should be agile, adaptive and aligned. We have argued that an enterprise must be reconfigurable to become triple-A, which is equivalent to evolution in biological systems.

Reconfigurable enterprise systems are agile if they are able to adapt to changing environment. In this respect, a complex adaptive system (CAS) perspective is well suited for modeling structural and behavioral dynamics that are present in the CMEE. The reconfigurable enterprise systems as a CAS emerge from interactions among nodes in the network, which evolve over time, driven by node-level decision and rules, and environmental conditions.

It is not hard to imagine how this type of dynamism would also induce unanticipated changes in the structure of ecosystem in general as happens. For example, today paper costs tend to determine the costs in the publishing business (Krajewski and Ritzman 1996). This is connected to energy costs which tend to drive the cost of paper. Therefore increases in energy costs lead to an increase in paper costs and, thus, publication costs. These excessive costs are causing publishing firms to seek different ways to reach their customers in a “paperless” fashion. At the same time, quite independent of rising energy costs, rapid
breakthroughs in information and computer technology and access make it possible to reach many customers via electronic and Internet-based media (e-books, e-journals, etc.).

Thus, changes in computer and telecommunications technology are prompting publishers to consider different media. So that negative developments in energy sector trigger a need to search for different resources in paper business, which realizes advancements in information and computer technology. Consequently, this dynamism serves to information and computer industry by switching the customer profile of publishing industry to this sector.

The rapid pace of innovation in hardware and software technology is another example that creates a constant demand for newer and faster products and applications in CMEE. Being the first enterprise to market a new or better product can mean success for both the product and the enterprise. Even for many relatively commonplace items, enterprises continue to result in better, cheaper products with more desirable features. For example, a company that develops a new kind of computer chip to be used in many brands of computers can earn millions of dollars in sales until a competitor is able to improve on that design. This dynamism puts a greater emphasis on research of CMEE than is typical in most manufacturing enterprise systems.

Therefore, considering the dynamic nature of a CMEE, enterprises that are slow to respond to such spontaneous changes may find themselves being left behind their competitors. And, often, enterprises that have become complacent in a stable market condition or long-term contracts may find themselves in this type of situation. Systems that turn over quickly stand a
better chance of exposing weak members and, thus, gaining higher efficiency than systems which are artificially bound by long-term relationship.

6.7. Adaptivity

Bureaucratic and monolithic organizations are ill suited to creating global teams or responding to a myriad of complex technological challenges on short notice. Although it is true that many individual enterprises may obey the deterministic selection process (Choi and Hartley 1996), the organization of the overall CMEE emerges through the natural process of order and spontaneity. In other words, all computer manufacturing enterprises operate according to self-interest and to promote their own fitness criteria and are, thus, governed by the invisible hand, which brings order and spontaneity over a course of time. What that means for individuals is that they need to constantly observe what emerges from a CMEE and make adjustments to organizational goals and supporting infrastructure.

Further, they should realize that it is quite normal for them to behave in a deterministic fashion based on few salient rules and performance measures. The key is to stay fit and agile and be willing to make appropriate adjustments in the face of changing environment and not be apologetic about making structural changes over a course of time (e.g., Hewlett-Packard decentralizing and then centralizing only to go back to a decentralized structure).

Given the short product life of many new electronic products, manufacturing processes and technologies computer manufacturing enterprises should be designed to be flexible and easy
to change. Enterprises that adjust goals and infrastructure quickly, according to the changes in their customers, suppliers, and/or competitors, survive longer in their CMEE than enterprises that adhere to predetermined, static goals and infrastructure and are slow to change. In those cases where a particular agent or enterprise fails to make appropriate adaptation and defeated to environmental disturbances, a hole is created in the ecosystem. In biological systems through a cascade of adaptation a new agent fills the hole.

It appears that enterprise ecosystem, like nature, has the similar characteristic, adaptation works in a very similar way and introduces a new enterprise to ecosystem. Given that the niche defines the agent, it should not be surprising that the new agent / enterprise tends to be very similar to the old one even though the two agents / enterprises may be of different genetic or historical backgrounds. This process, called convergence in evolutionary biology, seems to be active in enterprise ecosystem as well.

6.8. Rugged Landscape

A computer manufacturing related product is a function of CMEE attributes. These attributes combine in some manner to form a fitness or goodness value to the product. For example, a computer may be judged by its cost, its processor speed, its quality of graphics and sound, and its reliability. In most cases, making the product or offering a service better means attending to these underlying features. A key question is whether or not these features can be attended in an independent fashion. For example, in a desktop computer, one cannot completely decouple the quality of the computer from the quality of the processor: audio and
graphic cards affect the video quality, which, in turn, requires a better screen and compatible drivers and software and impacts customer satisfaction during playing a media.

More generally, we cannot assume that the overall product or service will improve if all elements of the contributing system improve — there may be inter-dependencies that overwhelm locally optimal contributions. For instance, a CMEE in which member enterprises seek to optimize their own local costs may in fact create wildly oscillating system behaviors (Forrester 1958; Sterman 1989) called the “Bullwhip effect” that ends up aggravating the system costs (Lee, Padmanabhan et al. 1997; Lee, Padmanabhan et al. 1997).

One way in which firms are attempting to deal with inter-dependencies is by moving toward modular design (Fleischer and Liker 1997; Meyer and Lehnerd 1997). Users of a modular design recognize that it is sometimes possible to develop an architecture, whereby contributing components are relatively autonomous. Therefore, they can be optimized independently and allow the emergence of high dimensionality. Such high dimensionality greatly cuts down on the coordination costs within the system. For example, a computer manufacturer may contract out the design of a key component; it will simply ensure adherence to an overall design by clearly specifying functional requirements, constraints (size, cost, time), and interface requirements (power supply, information connection, etc.).

Experience has shown that such modular designs require much more creative design effort up front, as compared to conventional (non-modular) design approaches. Modular design is one exemplary approach that may lead to success in such a situation. Theoretically speaking,
modularization of work tasks (e.g., design or manufacturing) reduces the number of peaks in the rugged landscape. Here, many small peaks are combined into a few large peaks, creating a condition more conducive to optimizing the overall system. Modularization of tasks will decrease overall inter-dependencies among firms in a supply network, and, thus, offer a higher efficiency when optimizing the overall system.

6.9. Co-evolution

Computer manufacturing enterprises react to their environment and thereby create their own ecosystem. Typically, significant dynamism exists in the CMEE, which necessitates a constant adaptation of the computer manufacturing enterprises. However, the ecosystem is highly rugged, making the co-evolution difficult. The individual enterprises constantly observe what emerges from the CMEE and adjust their organizational goals and supporting infrastructure. Therefore, CMEE management plays a critical role in making the connection between individual enterprises evolve in a coherent manner.

6.10. Quasi-equilibrium

Computer manufacturing enterprises generally maintain a stable and prevalent configuration in response to external disturbances. However, they can undergo radical changes when they faced with unpredicted rapid changes in the CMEE. At such a point, a small event can change balance of power and can trigger a cascade of changes that eventually can lead to a
system-wide redefinition or reconfiguration such that some enterprises survive and dominate
the environment, whereas some of them extinct.

In actuality, it is hard to identify the small event that eventually triggers a massive change
(by definition the small event, such as flapping of butterfly wings, is hard to identify). However, we can always identify and observe the eventual change itself. For example a
system that becomes destabilized due to qualitative changes (by either the economic or
technological advancements) can undergo catastrophic, unpredictable changes that lead it
into a different patterned behavior or a new reconfiguration.

A well known example of such a massive change in CMEE is IBM, which has a long and
varied history that goes as far back as the late nineteenth century. IBM has been responsible
for many firsts in the computer field, was the leader of its sector until 1990s. In the 1990s,
the computing business underwent a major revolution. Even though it had played an
important role in pioneering the personal computing industry, IBM began to lose money. Many competitors enter the computer manufacturing sector which reduces the influence of
IBM in this field and CMEE undergone a radical structural change. But, in sum, CMEE
maintains its stable and prevalent configuration after a while with its new reconfiguration.

6.11. Non-linearity

The interactions among enterprises involve the transmission of knowledge and materials that
often affect the behavior of the enterprises. Those interactions among the enterprises are
diverse, variable, and non-linear. The idea of order and spontaneity leads one to consider that self-organization of the computer manufacturing enterprises does not occur in a linear or sequential manner. It occurs in parallel.

For example, CMEE has a wide geographic distribution. Customers can initiate transactions at any time with little or no regard for existing configuration of a computer system, thus contributing to a dynamic and noisy network character. The characteristics of a network tend to drift as workloads and configuration change, producing a non-stationary behavior. The coordination protocols attempt to arbitrate among entities with resource conflicts. Arbitration is not perfect, however; hence, over- and under-corrections contribute to the nonlinear character of the network.

In theory it is possible to have a sequential and isolated organization of the CMEE by minimizing the scope of study and reducing the number of constraints. However, as discussed above, the CMEE is typically organized in a networked and inter-dependent way, where events occur nonlinearly. Over simplification of the system fall short of reality.

Sub-clusters exist within the CMEE which use several related sources and spontaneously create a whole series of overlapping complex structures. These complex structures may overlap and draw resources from more than one source, for instance, a mining company can supply raw material to telecommunications industry as well as the toy industry. Since CMEE operates non-linearly, it is not possible for one enterprise to control its operation in deterministic fashion. This is an important realization for many enterprises whose
management has developed an unfounded belief that the goal of an enterprise should be to gain control over the entire system. However, today’s experts of CASs (Choi, Dooley et al. 2001; Surana, Kumara et al. 2005) argue that in contrast, the goal should be to develop a strategy to decide how much to control and how much to let emerge in the system.

6.12. Non-random Future

Past is often not a good predictor of the future of a business market in which technologies redefine the rules of the game. CMEE is a market, which is characterized by technological innovations, has unpredictable features, meaning that it is risky to make exact predictions of the computer manufacturing enterprises future development.

Even though an exact prediction of future behavior is difficult in this system, often archetypal behavior patterns can be recognized. In fact the actual structure of a CMEE will depend on local choices and how these choices are combined to create a probabilistic entity. Moreover, characteristics of CMEE at any one time seem to be affected by their past histories. Therefore chances are good that if one were to replicate the past histories and some of the same behavioral tendencies would tend to emerge. Also network connectivity and centrality is another factor.

For instance, in the high-tech industry, a small shift in the consumer market can become amplified and has a huge impact on equipment manufacturers. Therefore a higher rate of bankruptcy is observed among equipment manufacturers who are distant to customer market
in the network and therefore unaware of changes in this market segment, compared to for example, DRAM chip manufacturers, who are closer to the consumer market and therefore more quickly anticipate and synchronize with the changes (Fine 1998).

6.13. Difficulty to Determine Boundaries

Operationally, the boundaries of CMEE depend on the chosen scale of analysis, e.g., they can simply be taken as computer and peripheral manufacturers, or larger joint set of manufacturer such as computer and peripheral manufacturers, semiconductor and other electronic component manufacturers, navigational, measuring, electro-medical, and control instruments manufacturers, magnetic and optical media manufacturers and software enterprises, etc.

In a complex adaptive system, boundaries are emergent and temporary. A particular domain, or structure or subsystem, may seem to spontaneously appear, persists for a long period and then fades away. For example, particular organizations or industries can be seen as emergent domains that are apparently self-sustaining and separate from other organizations or industries. Therefore, CMEE boundaries can change as a result of including or excluding particular agents and by adding or eliminating connections among agents, thereby changing the underlying pattern of interaction in the virtual world.

A CMEE is highly structured and heterogeneous system where, the components of complex CMEE are complex systems as well. For instance, in the CMEE structure, at the upper level we observe a computer manufacturing enterprise as a single entity. However, at the lower level, we observe entities, such as people, that exist inside this computer manufacturing enterprise.

The level taken to make sense of a complex system depends upon the accuracy required or the practically achievable. For example, organizations are very difficult to understand in terms of individuals so they are often described in upper level as coherent systems in themselves with the whole only being assumed to exist. Therefore, considering the required and achievable accuracy of this dissertation we study the evolution and organization of CMEE in enterprise-enterprise and enterprise-customer level.
Chapter 7

Modeling of CMEE

A computer simulation/model is a computer program that attempts to simulate an abstract model of a particular system. Computer simulations are a useful part of mathematical modeling of many natural systems in physics (computational physics), astrophysics, chemistry and biology, human systems in economics, psychology, social science, and engineering. Simulations are used to explore and gain new insights into new technology, and to estimate the performance of systems too complex for analytical solutions.

Goldsman, Nance et al. (2009) state that the history of simulation modeling can be written from different perspectives, such as uses of simulation (analysis, training, research); types of simulation models (discrete-event, continuous, combined discrete-continuous); simulation programming languages or environments (GPSS, SIMSCRIPT, SIMULA, SLAM, Arena, AutoMod, Simio); and application domains or communities of interest (communications, manufacturing, military, transportation).

First example of simulation modeling considered being date back to 1777; it is The Buffon “needle experiment” which was restudied by Laplace in 1812. About a century after Laplace’s study, William Sealy Gosset (Pearson, Gosset et al. 1990) published certain major statistical results, by using pseudonym and no proprietary data. These results were published under the
pseudonym “Student” beginning in 1908 with a paper formulating what is now known as Student’s t-distribution. Because Gosset had incomplete analytical results, he used a crude form of manual simulation to verify his conjecture about the exact form of the probability density function for Student’s t-distribution. In the mid-1940s two major developments set the stage for the rapid growth of the field of simulation during the period of 1945 - 1970; the construction of the first general-purpose electronic computers and several researchers’ use of the Monte Carlo method on electronic computers. And later during 1970 - 1981 the field of simulation developed enhanced modeling tools and analytical tools.

The role of simulation becomes more important and critical with developing technologies and increasing complexity during 1990s. At present, as the economic, social and environmental problems confronting all mankind become increasingly critical and interrelated, engineers have an extraordinary opportunity to take the lead in synthesizing effective solutions that draw on all areas of specialized technical knowledge and the sheer complexity of these problems dictates that simulation modeling will be an essential tool for crafting good solutions.

In general, advantage of using computer simulation is that it is necessary to think through one’s basic assumptions very clearly in order to create a useful simulation model. Every relationship to be modeled has to be specified exactly. Every parameter has to be given a value, otherwise it will be impossible to run the simulation. This discipline also means that the model is potentially open to inspection by other researchers, in all its detail.
In contrast to the inductive methodology of collecting data and then building models that describe and summarize those data, simulation approach starts from a more deductive perspective. A model is created, calibrated from whatever data is available and then used to derive testable propositions and relationships. This approach places much lower demands on the data, while the models can truly reflect the complex nature of systems. However simulations of complex enterprise ecosystem involve the estimation of many parameters, and adequate data for making the estimates can be difficult to come by. Another benefit of simulation is that, in some circumstances, it can give insights into the emergence of macro level phenomena from micro level actions.

There are two fundamental types of simulation models: deterministic and stochastic. Deterministic models always produce the same outputs when they are repeatedly run with identical inputs. Stochastic models can produce different outputs when they are repeatedly run with identical inputs. Stochastic models can produce different outputs because they include agent behaviors or environmental responses based on random or probabilistic elements. By definition, such models need to be executed many times to produce valid general results.

Table 7.1 describes six different types of computer simulations related to emergent system response prediction (Allen 2011). Detailed comparisons of these techniques can be found in the literature.
Table 7.1 Selected types of simulation and how they operate (Allen 2011)

<table>
<thead>
<tr>
<th>Simulation type</th>
<th>Engine</th>
<th>Relevance and data sources</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agent-based</td>
<td>Agent rule iteration and Monte Carlo</td>
<td>Particularly relevant for studying incentives and restrictions and based on low-level data</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Discrete event simulation</td>
<td>Event controller and Monte Carlo</td>
<td>Particularly relevant for studying production systems and based on low-level data</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Forecasting</td>
<td>Empirical modeling including least squares</td>
<td>Particularly useful for predicting new emergent properties based on high-level data</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Markov chain models</td>
<td>Linear algebra</td>
<td>Relatively simple and transparent and based on expert opinion and/or high-level data</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Systems dynamics models</td>
<td>Differential equation numerical solvers</td>
<td>Particularly relevant for studying the impacts of decisions based on expert opinion</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Physics-based</td>
<td>Finite element methods (FEM) numerical solvers</td>
<td>Particularly relevant for designing engineered products and based on low-level data</td>
</tr>
</tbody>
</table>

The benefits of agent-based modeling (ABM) over other modeling techniques can be captured in three statements (Bonabeau 2002): (i) agent-based modeling captures emergent phenomena; (ii) agent-based modeling provides a natural description of a system; and (iii) agent-based modeling is flexible. It is clear, however, that the ability of ABM to deal with emergent phenomena is what drives the other benefits. And three ideas central to agent-based models can be summarized as emergence, social agents as objects, and complexity.
An enterprise ecosystem has always been too complex for us to adequately model and the complexity of the multidisciplinary and interdisciplinary problems in this field necessitates simulation modeling. In other terms assumptions of perfect markets, homogeneous agents, and long-run equilibrium that made the problems analytically and computationally tractable are not as applicable as they once were. Therefore, like any CAS, study of enterprise systems, should involve a proper balance of simulation and theory. Traditional modeling methodologies are often considered inadequate to explain the complexity of enterprise ecosystems. Therefore, system dynamics based and agent based simulation models can be extensively used to make theoretical investigations of these systems feasible and to support decision making in real world.

Measuring and knowing in the business world require business leaders to handle more uncertainty than ever before. The need to capture the behavior of the large number of complex (complexity) interacting components (organizational agents) of business world creating surprising results (emergence) directly calls for agent simulation. As history shows many researchers have developed agent-based modeling systems to answer questions about life and self-organization.

Today, managers of enterprises and business people should do the same for their questions. Previous historical events can be analyzed to determine not only what did happen but also what might have happened. Uncovering unexpected issues through trial and error in the real world and collecting good information can be quite costly. Maintaining deep databases takes real investments. The quality of information is never perfect and never will be. However using
simulation models help us to speed up organizational learning, avoid problems before they happen, and recognize opportunities before they are obvious to everyone.

Moreover, using an agent-based approach, evolution of enterprise systems can be simulated on the computer. It is even possible to carry out experiments on artificial environments where each enterprise system has a chance of experiencing to emerge, evolve, and extinct which would be quite impossible to perform or observe in real time on enterprise ecosystems.

In the light of this understanding, we developed an agent based simulation model to understand the dynamics of CMEE. We used a special purpose simulation language and modeling environment (NetLogo) to simulate enterprise ecosystem. The enterprises constructed using collections of condition-action rules to be able to “perceive” and “react” to their situation, to pursue the goals they are given, and to interact with other enterprises, for example by sending and receiving materials via their supply networks.

The remainder of this chapter explains the concept of agent based simulation modeling in more detail. In the next chapter, in simulation framework section, agent based simulation model of the CMEE will be explained further by using agent based terminology, showing how the agent based simulation methodology is appropriate for analyzing enterprise systems phenomena that are inherently complex, emergent, and self-organizing.
7.1. Agent Based Simulation Modeling

An agent-based simulation (ABS) modeling is a class of computational models for simulating the actions and interactions of autonomous agents with a view to assessing their effects on the system as a whole. It combines elements of game theory, complex systems, emergence, computational sociology, multi-agent systems, and evolutionary programming. The models simulate the simultaneous operations and interactions of multiple agents, in an attempt to re-create and predict the appearance of complex phenomena.

Agent based models were derived from work in a sub-area of artificial intelligence called distributed artificial intelligence (DAI) (Luger 2009). DAI aimed to solve problems by dividing them amongst a number of programs or agents, each with its own particular type of knowledge or expertise. In combination, the collection of agents would be better at finding solutions than any one agent working on its own. While DAI is primarily concerned with engineering effective solutions to real world problems, it was soon noticed that the technology of interacting intelligent agents could be applied to other fields such as modeling social phenomena, with each agent representing one individual or organizational actor.

Agent based simulations usually consist of a number of agents communicating via messages passed between them through an environment. All the agents and the environment are represented within a computer program. Often the environment is modeled as a two-dimensional space and each agent is positioned in a different location. In some models, the agents are free to travel thorough the space while in others they are fixed. So, most agent-
based models are composed of: (1) numerous agents specified at various scales; (2) decision-making heuristics; (3) learning rules or adaptive processes; (4) an interaction topology; and (5) a non-agent environment.

7.2. Agent

An agent based model consists of a number of software objects, the core units of simulation, “agents”, interacting within a virtual environment (Terano 2005). An agent is identifiable, a discrete individual with a set of characteristics and rules governing its behaviors and decision-making capability. Agents have sets of decision rules that govern their behaviors. These rules allow agents to interact with and communicate with other agents as well as to respond to their environments. These rules can provide agents with responsive capabilities on a variety of levels from simple reactions to complex decision-making. An agent is flexible, and has the ability to learn and adapt its behaviors over time based on experience. There are two techniques commonly used for this: neural networks and evolutionary algorithms such as the genetic algorithms. Wooldridge and Jennings (1995) stated that agents generally have the following properties:

- Autonomy: Agents control their own actions
- Social ability: Agents interact with other agents through a computer language
- Reactivity: Agents can perceive their environment and respond to it.
- Pro-activity: As well as reacting their environment, agents are also able to undertake goal-directed actions.
In general, agent behaviors follow three overall steps. First, agents evaluate their current state and then determine what they need to do at the current moment. Second, agents execute the actions that they have chosen. Third, agents evaluate the results of their actions and adjust their rules based on the results.

### 7.3. Agent Attributes

In general agents have the following attributes:

#### 7.3.1. Knowledge and Belief

Agents need to base their actions on what they know about their environment. Some of the information they have may be incorrect, as a result of faulty perception, faulty inference or incomplete knowledge. Such erroneous information is different from true knowledge and called the agents’ belief.

#### 7.3.2. Inference

Given a set of beliefs, agents may be able to infer further information from them. For example, believing that agent B has recently eaten some food, agent A could infer that the place to find food is near where agent B was located. However, this inference may be
wrong, if the agent B consumed all the food in that location. Therefore, agent A should be able to learn from its mistakes.

7.3.3. Social Relationship

Some agents may be capable of learning about the interrelationships between other agents in their world. On the basis of such data, agents may be able to put together a picture of the social relationship in their environment. Agents may learn other aspects of their environment such as geography. In these types of agent based simulation models, the model is built by the agents themselves while the simulation runs.

7.3.4. Goals

Since agents are built to be autonomous and opportunistic, they need to be driven by a need to satisfy some internal goal such as survival. Survival may in turn require the satisfaction of subsidiary goals, such as acquiring energy and avoiding lethal dangers. The main difficulty in designing goals is how to make agents to define their own subgoals relevant to the situation at hand. Further there can be difficulties in deciding how to manage several goals which may be of differing importance and relevance and which may possibly conflict. The solution to these problems is often the responsibility of a planner module built into the agent.
7.3.5. Planning

An agent needs to have some way of determining what behavior is likely to lead to the satisfaction of its goals. This may be very straightforward. An agent may be programmed to move away if it finds itself adjacent to a stronger aggressive neighbor. Such simple condition-action rules can be very powerful when several are used in combination, but often it is desirable for agents to do more complex planning. Planning involves working backwards from a desired goal state, inferring what action would lead to that goal, what state would be required before that action can be carried out, what action is needed to arrive at that state, and such, until the agent gets back to the current situation. Sophisticated planners can be built but it has argued that the real actions are usually driven by routine reactions to a situation rather than elaborately calculated plans.

7.3.6. Language

All agent based simulation models include some form of interaction between agents or interaction between agents and the environment in which they exist. The interaction may involve the passing of information from one agent to another, the negotiation of contracts, or even one agent feeding on another such as predator-prey models. In some models interaction may only convey factual or non-intentional meaning. In most models intention of interaction is communication with other agents. Such communications need to be modeled by a language. Developing specialized computer languages for agent communication has considerable difficulties and it can be avoided passing messages directly between ages.
7.3.7. **Knowledge Representation**

In order to construct models, an agent needs some way to represent its beliefs. Techniques for doing this have been studied in artificial intelligence field under knowledge representation. One general useful approach is to use predicate logic to store declarative statements. Another approach, which can be used alone or in conjunction with logic, is based on semantic networks in which objects and their attributes are related together, often as a hierarchy.

7.4. **Environment**

Agents are almost always modeled as operating within some social environment consisting of a network of interactions with other agents. Sometimes, however, it is also useful to model them within a physical environment that imposes constraints on the agents' location. What constitutes an environment depends on what is being modeled. In many models, agents are able to move around the environment; in others they are stable. Once agents are positioned within an environment, they need to perceive their local neighborhood and interact with their environment. Usually, communication between agents is routed through the environment, which forwards messages to appropriate recipients. The usual assumption is that nearby agents are more likely to interact or are better able to influence each other than those farther apart.
Chapter 8

Simulation Framework

An enterprise ecosystem can be considered as an environment consisting of enterprise and customer interactions in that environment; and collaboration links that form or dissolve among enterprises. In that environment each enterprise has certain strengths and weaknesses relative to other organizations in the industry. Also each enterprise has different level of understanding of triple-A measures which provide the fundamental basis for the provision of added value.

In the light of this basic understanding, an enterprise ecosystem can be represented as a complex adaptive system which consists of interacting enterprises in the enterprise ecosystem. Furthermore, agent based simulation models can be used to make theoretical investigations of CMEE feasible and to support decision making in real enterprise ecosystems. In our simulation model computer manufacturing enterprises correspond to agents communicating in the virtual CMEE and collaborations between enterprises, material flow and information flow from one enterprise to another, correspond to links. Furthermore computer manufacturing enterprise ecosystem corresponds to the environment.

The enterprise agents have a degree of autonomy, to react to and act on CMEE and on other computer manufacturing enterprise agents, and have goals that they aim to satisfy. In CMEE,
collaboration is an essential mechanism to integrate and propagate information distributed pervasively among computer manufacturing enterprise agents. In such a model, the virtual computer manufacturing enterprises can have a one-to-one correspondence with real-world computer manufacturing enterprises, while the interactions between the virtual world computer manufacturing enterprises can likewise correspond to the interactions between the real-world computer manufacturing enterprises. With such a model, it is possible to initialize the virtual world to a preset arrangement and then let the model run and observe its behavior. Specifically, emergent patterns of action may become apparent from observing the simulation.

The CMEE, in which enterprises reside and interact, can be characterized by a static and a dynamic component. The static component consists of environmental rules and environmental fitness threshold level. Environmental rules represent the rules of behavior that characterize and define an enterprise. For example, in order to satisfy Ashby’s Law (Ashby 1964; Tharumarajah 2003), an enterprise should be able to handle successfully both predictable and unforeseeable changes. However enterprises which are not scalable or flexible enough will go extinct in the ecosystem eventually. The extinction will lead to atrophy of collaboration links and isolation and death of the enterprise.

The enterprise survival or extinction in the enterprise ecosystem is determined by environmental fitness threshold values. For example environmental fitness threshold value for survival represents the necessary fitness level that each enterprise in the environment must sustain in order to “live” in the enterprise ecosystem. In that sense each enterprise has a
fitness value and must be fit for its tasks, in order to be a viable member in the environment. For example, every enterprise must sustain some level of financial viability, if it falls below this level, the enterprise may go bankrupt and die in the simulation.

The dynamic component of the environment includes satisfying customer needs, time-varying demand and price in that environment and number of enterprises that exist in the environment. In this environment, every agent has a set of strategies that they use in making decisions to achieve their objectives under some constraints set by the environmental rules. Generally, enterprises make two types of decisions; with whom to link in the environment, and how to set their attribute levels.

In the multi-agent simulation model, enterprises encounter both familiar and unfamiliar situations and learn to adapt to those environmental conditions based on prior success and failure. Such adaptation can incorporate a wide variety of behaviors and intelligence. Single loop learning – the ability to self-diagnose and correct a situation when there is a deviation from predefined operating norms – is sufficient if the enterprise faces only familiar situations. In order to deal with new and unfamiliar situations in a timely manner, i.e., to possess “agility”, the enterprise must engage in double-loop learning, they should self-diagnose forthcoming changes and take actions to modify operating norms if needed.
8.1. Multi-Agent Framework of CMEE

Dynamics of any ecosystem become complicated due to the presence of multiple performance measures and complex interactions. Also the variations in each system require a better knowledge of the structure and dynamics of those systems. Multi-agent framework of CMEE allows observation of decision making process, information flow, collaboration and network structure of CMEE. Multi-agency provides the ability to observe materials flow, distribution, inventory control, information exchange, supplier reliability, number of suppliers, demand forecast mechanisms, and flexibility to change commitments.

8.2. Agent Based Simulation Model of CMEE

We develop an agent based simulation model as shown in Figure 8.1 to understand the dynamics of CMEE. We use the special purpose simulation modeling environment NetLogo (Wilensky 1999) to simulate the enterprise ecosystem. The enterprises are constructed using collections of condition-action rules to be able to “perceive” and “react” to changes in their environment to pursue their goals and to interact with other enterprises, for example by selling and buying goods via supply-demand links.

CMEE agents consist of the customers and suppliers (main supplier or Tier 0, Tier 1, and raw material supplier or Tier 2). Some common attributes used to represent computer manufacturing enterprises competing within markets include resources, size, decision response time, profit / volume / market share targets, and risk tolerance.
Some common behaviors include daily / weekly / monthly operations, annual planning, and unexpected event response. Each iteration of multi-agent simulation of CMEE, CMEE agents execute behaviors:

1. The customer places an order with the main supplier.
2. The main supplier fills the order immediately from its respective inventory if it has enough inventories in stock (if the main supplier runs out of stock, in some cases customer is returned and others the customer’s order is placed on backorder and filled when stock is replenished).

3. The main supplier receives a shipment from the Tier 1 in response to previous orders. The main supplier then decides how much to order from the Tier 1 based on an “ordering rule.” The ordering decision is based in part on future demand. The main supplier estimates future customer demand using a “demand forecasting” rule. The main supplier then orders items from the Tier 1 to cover expected demand and any shortages relative to explicit inventory or pipeline goals.

4. Similarly, each Tier 1 receives a shipment from the raw material supplier, forecasts future demand by the main supplier, decides on how much to put into new production and places an order with the raw material supplier.

This process continues in the supply chain with each new interaction between customer-main supplier, main supplier-Tier 1, Tier 1-raw material supplier.

8.3. Agents of CMEE

An agent based model consists of a number of software objects called agents, the core units of simulation, interacting within a virtual environment (Terano, Arai et al. 2005). In our simulation model, computer manufacturing enterprises (CMEs), their suppliers and
customers are agents communicating in the virtual CMEE. Each agent has a degree of autonomy and has goals that it aims to satisfy. It also reacts to and acts on CMEE and on other horizontally or vertically interacted agents. Each agent specializes according to its intended role in the CMEE.

There are two types of main agents in the simulation model: customer and supplier. Customer agents are divided into 3 groups according to their shopping preferences:

1. Customers who buy the product provided by the most reliable supplier in their area of search.
2. Customers who shop according to the popularity of the product provider in their area of search.
3. Customers who buy the product offered with lowest market price in their area of search.

Customer agents use a modified particle swarm optimization (PSO) algorithm of Stonedahl and Wilensky (2008) to search through the enterprise ecosystem and find their preferred main supplier. PSO is a search / optimization technique in the field of machine learning. In PSO optimization technique, main goal is to find values for \( x \) and \( y \), such that fitness function \( f(x, y) \) is maximized. The fitness function determines how good the current position in space is for each customer. One approach we applied in the beginning of our research is the random search technique, where \( x \) and \( y \) values are randomly selected and the largest \( f(x, y) \) found in the population is recorded. This search technique is very easy to implement, however it is not efficient for many search spaces. Therefore we replaced it with modified PSO which is more
"intelligent" in its search process. Customers are placed in the search space, and move through the space according to rules that take into account each customer's personal knowledge and the global "swarm's" knowledge. Through their movement, customers discover particularly high values for the fitness function.

PSO algorithm that we used in the multi-agent CMEE simulation model can be summarized as follows: each customer agent has a position \((x_{\text{cor}}, y_{\text{cor}})\) in the search space and a velocity \((v_x, v_y)\) at which it is moving through that space. Customer agents have a certain amount of inertia, which keeps them moving in the current direction. They also have acceleration (change in velocity), which depends on two main things:

- **Attraction-to-personal-best**: Each customer agent is attracted toward the best location that it has personally found (personal best) previously in its history.

- **Attraction-to-global-best**: Each customer agent is attracted toward the best location that any customer agent has ever found (global best) in the search space.

How fast customer agents move in each direction (strength of attraction force) depends on the attraction-to-personal-best and attraction-to-global-best parameters. As customer agents move farther away from the "best" locations, the force of attraction grows stronger. There is also a random factor about how much the customer agent is pulled toward each of different locations other than the "best" locations.

Supplier agents are divided into three specific groups that are vertically integrated: Tier 0 (supplier of main interest), Tier 1 and Tier 2 according to a hierarchical structure of who
supplies who. All supplier agents share specific common attributes, including the inventory, demand, product price, production rate, budget, energy, and fitness value.

Supplier agents use the concept of adaptation from complex adaptive systems. During a simulation run they might be in three different states: adapted, resistant or neutral. These three measures are explained in more detail in the following sections. Supplier agents also use the concept of cooperation from evolutionary biology. In it, supplier agents compete for CMEE resources. Supplier agents that are more successful in competition grow faster and open new branches in new locations (reproduce). In the model two kinds of competition methods being greedy and being cooperative are implemented. These two different strategies are examined when supplier agents competing against each other within CMEE that evolves over time.

Network of supplier agents is also another main interest of this dissertation. We use the network theoretic concepts and investigate the resulting CMEE supply network structure. At each step, a newcomer (new supplier agent) is added to the network. The newcomer might be a totally new member or a growing member that is already participating in the CMEE, which reaches the reproduction threshold. The newcomer picks an existing enterprise to connect to randomly or using preferential attachment. In preferential attachment an enterprise's chance of being selected is directly proportional to the number of connections it already has, or its degree.
We also explore the formation of CMEE network that may result in the small world phenomenon -- the concept that a social entity is only a couple of connections away from any other social entity in a network. Networks with short average path lengths and high clustering coefficients are considered small world networks. Therefore, to identify small worlds, the average path length and clustering coefficient of the CMEE network are calculated. We also investigate the CMEE network giant component size. In a network, a component is a group of nodes that are all connected to each other, directly or indirectly. The component with the most members at any given point in time is the giant component. So, if a network has a "giant component", almost every node is reachable from almost every other node. Therefore the size of the giant component is one of the network measures that show the degree of connectivity in the CMEE network.

In CMEE, an agent, regardless of its type, has the following set of characteristics at a given time instant:

**Agent Attributes:** Attributes characterize an agent’s state at a given instant of time. Current inventory, current financial position, current market position, different costs associated with production are some of the attributes that characterize an agent’s state. Dynamic attributes change over time either as the result of internally triggered events such as material transfers from work-in-process inventory to finished-goods inventory or as a result of interactions with other agents such as receipt of an order, shipment of an order and payment for an order.
Agent Knowledge: Generally, an agent’s knowledge consists of a global view and a local view. Global view requires the knowledge of states and actions of other agents. In our simulation model, customer agents have the global view in terms of being aware of brand preferences and favorite products of other customer agents. This is a realistic assumption considering that customers in today’s world have combined information and social networks such as television advertisements and social networking services. However, supplier agents have a local view. Therefore, each supplier agent has incomplete information of the state, actions, current decisions, future plans and past performance (with the exception of published reports of suppliers to public) of other agents. As a result, each supplier agent’s knowledge is limited to the information about its immediate neighbors. A supplier agent decides its current and future actions according to its local neighbors (link neighbors) and market averages.

Agent Interactions: Interactions define an agent’s relationship with other agents and with its environment. In CMEE, collaboration is an essential mechanism to integrate and propagate information distributed pervasively among agents. In such a model, the virtual enterprises and customers can have a one-to-one correspondence with real-world enterprises and customers. In the same manner, the interactions between the agents in the virtual world, CMEE, can correspond to the interactions between the real-world markets. With such a model, emergent patterns of action (e.g., “enterprises and customers”) may become apparent from observing the simulation.

In the simulation model, customer agents have a global view and therefore interaction among customer agents is not the point of interest to this research. However, supplier agents have a
local view which introduces significant importance to their local interactions. A supplier agent’s state in the ecosystem is affected by its neighbor supplier agent’s decisions. As a result of these local interactions, the ecosystem evolves, changes, and introduces unexpected vulnerabilities into the ecosystem. Supplier agents experience both vertical and horizontal interactions.

The following are the horizontal interactions:

1. Customer to customer
2. Tier 0 to Tier 0
3. Tier 1 to Tier 1
4. Tier 2 to Tier 2

The following are the vertical interactions:

1. Customer and Tier 0
2. Tier 0 and Tier 1
3. Tier 1 and Tier 2

Agent Behavior and Learning: In the simulation model, enterprises encounter both familiar and unfamiliar situations and learn to adapt to those environmental conditions based on prior success and failure. Such adaptation can incorporate a wide variety of behaviors and intelligence. Single loop learning – the ability to self-diagnose and correct a situation when there is a deviation from predefined operating norms – is sufficient if the enterprise faces
only familiar situations. In order to deal with new and unfamiliar situations in a timely manner, i.e., to possess “agility”, the enterprise must engage in double-loop learning; they should self-diagnose forthcoming changes and take actions to modify operating norms if needed.

In our simulation model, in order to secure their positions, suppliers in different tiers keep track of their number of vertically and horizontally integrated neighbors; if the number of their links is below a certain threshold they immediately look for new connections. Also suppliers drop their weakly connected neighbors and try to link with stronger candidate collaborators. Here they use the revenue of each company as the selection measure.

Supplier agents also keep track of the technological changes in the market. An enterprise might choose to adapt to changes in the market or become resistant to changes. Therefore, an enterprise might be in either one of the three states: adaptive, resistant, or neutral.

- Adaptive: Company follows and introduces technological changes.
- Resistant: Company is resistant to respond to any change in the market.
- Neutral: Company track changes in the market and ready to take action but not taking any actions yet.

According to their predefined strategies, each supplier might either try to align with other companies in the system by changing its product price closer to market average, or if the
company is not interested in aligning with other companies in the market and prefers to compete greedily, it decreases its product price below market average to sell more products.

8.4. Agent Ecosystem

In our simulation model, the CMEE can be considered as consisting of (1) an environment consisting of enterprises and customers; (2) enterprises which correspond to supplier agents; (3) customers who correspond to customer agents; and (4) collaboration links, which provide information flow from one enterprise to another, that form or dissolve among those enterprises according to environmental rules and agent decisions.

In this agent ecosystem, environmental disturbances force enterprises to quickly adapt to changes. This adaptation is required to secure or improve the current position of an enterprise in the market. Environmental disturbances can come from three primary sources: (1) customer demand for individualized and customized products at lower costs; (2) predictable changes created by such factors as technology advances, social trends, and environmental concerns; and (3) unforeseeable changes such as natural calamities, war threats, and technological breakthroughs.

**Agent Ecosystem Functions:** Functions are a set of equations that determine which rules are to be used based on domain knowledge, current state, and priorities of an agent. For example, in the ecosystem, an enterprise has its own neighbors and its potential neighbors. In each
simulation iteration, every enterprise in the system tries to eliminate its weakest link neighbor and tries to link the strongest potential neighbor according to a function considering the financial criteria.

**Agent Ecosystem Rules:** Rules define the event handling routines. Rules used in each iteration step depend on the function processed and the type of the agent. For example, in each iteration step, agents decide to become adaptive, resistant, or neutral according to an adaptation function. If an enterprise decides to become adaptive, the rules belonging to adaptive agents are applied to that enterprise. While some rules are simple if-else cases which use logic comparison or boolean variables mostly for decision making, other rules are more complicated in structure and contain iterative or recursive processes. Following are the rules that our simulation model uses.

- **Material Flow Rules:** Rules in this category model the delivery of goods by one agent to another. The processing semantics associated with material delivery dictate adjustments to inventories, demand and supplies of supplier agents. These supply-demand interactions between agents trigger information and cash flows. Also they affect the metabolism rate (energy level of a supplier to survive in the ecosystem) and production rate of interacted suppliers. In our simulation model, material transports happen directly between supplier agents of different tier levels, and from Tier 0 supplier agents to customer agents.
• **Information Flow Rules:** Rules in this category model the exchange of information between agents, such as request of an item, informing about fulfilled or rejected orders, and payment received information for an order.

• **Cash Flow Rules:** Rules in this category handle the cash related events, such as updating budget of supplier agent and current purchasing power of customer agents.

• **Agent Based Rules:** Rules in this category regulate the inner dynamics and update the current state of each agent in the system accordingly. For example, each company tries to manage its inventories according to the following rule: if they have enough supply and demand they increase their production-rate; if demand decreases or there is not enough supply they decrease their production rate accordingly.

• **Global Rules:** Rules in this category regulate the system dynamics. For example, if the system loses balance with respect to the number of each type of enterprise in the system, the global rules force some of the enterprises, which belong to the supplier type that dominating the market, to change their type to the weakest supplier type. Another example is related to the environmental conditions. Environmental conditions may change randomly or according to the simulation clock or market performance of enterprises.

### 8.5. Network Structure of CMEE

Other agent interaction topologies, such as networks, allow an agent’s neighborhood to be defined more generally and flexibly. A network is a set of items, which we call vertices or

In this dissertation, the interactions of supplier and customer agents as well as the collaboration links make up the agent network of our simulation model. Supplier agents provide products, services, and information and add value to computer manufacturing enterprises, related industries and customers. Customer agents purchase products, set market trends and determine product popularities.

For enterprises, surviving and succeeding in the CMEE mean balancing a timely response to environmental disturbances against manufacturing and inventory costs. A computer manufacturing enterprise evolves in the uncertain rapidly changing enterprise environment
by adapting and synchronizing throughout internal and external linkages. In addition to the inherent complexity of this ecosystem, it is also hard to coordinate the actions of entities across organizational boundaries (vertical and horizontal interactions between tiers) to make them perform in a coherent manner.

In our study, the computer manufacturing enterprises, and also their Tier 1 and Tier 2 suppliers are treated as nodes. Supply-demand links, which contain the collaboration links and the information flow, play the role of links that are built from environmental thresholds in the enterprise ecosystem. Here, if a connection is the weakest link in its neighborhood, the enterprise who owns that link tries to replace the weakest link with one of the better candidates from the ecosystem, otherwise the connection is kept without a change. In this fashion, the enterprise network self-organizes and evolves until the extinction of a certain type of suppliers is observed. During the simulation run various properties of computer manufacturing enterprise network computed: they are density, giant component, average shortest path length, and clustering coefficient.
Chapter 9

Design and Analysis

In this chapter multiple pair-wise comparisons of different system design alternatives are conducted to determine the main and interactive effects of system variables on performance measures. Then in the following chapter, the results of these analyses are analyzed statistically. Moreover network theory is applied to analyze features and structure of the CMEE.

9.1. Controllable and Uncontrollable Variables

A controllable variable is a parameter that is varied during an experiment to study the effect it presumably has on a response. Uncontrollable variables that are inputs to the model are assumed to be external to the system. In our simulation, the controllable variables, which are set by the decision maker, and the uncontrollable variables that are observed are shown in Table 9.1.

9.2. State Variables / Attributes

In NetLogo, which is an integrated modeling environment, a variable can be a global variable, a turtle / agent variable, a patch variable, or a link variable. If a variable is a global variable, there is only one value for the variable, and every agent can access it.
### Table 9.1 Controllable and uncontrollable variables

<table>
<thead>
<tr>
<th>Controllable Variables</th>
<th>Uncontrollable Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial number of nodes</td>
<td>Number of customers that arrive in each iteration</td>
</tr>
<tr>
<td>Initial average node degree</td>
<td>Inter-arrival time of each customer</td>
</tr>
<tr>
<td>Initial adaptation size</td>
<td>Effects of customer purchase behavior on different supplier types</td>
</tr>
<tr>
<td>Environmental effects</td>
<td>Effects of environmental conditions on suppliers</td>
</tr>
<tr>
<td>Reproduction cost</td>
<td>Effects of environmental conditions on market attributes</td>
</tr>
<tr>
<td>Reproduction threshold</td>
<td>Effects of reproduction threshold on suppliers</td>
</tr>
<tr>
<td>Adaptation, lose adaptation, or resist</td>
<td>Supplier and customer volume that deviates with environmental conditions in the market</td>
</tr>
<tr>
<td>adaptation probabilities</td>
<td></td>
</tr>
<tr>
<td>Limited income threshold</td>
<td></td>
</tr>
<tr>
<td>Metabolism rate</td>
<td></td>
</tr>
<tr>
<td>Number of inventory to hold</td>
<td></td>
</tr>
<tr>
<td>Number of items to produce in each iteration</td>
<td></td>
</tr>
<tr>
<td>Number of supplies that are purchased from suppliers.</td>
<td></td>
</tr>
<tr>
<td>Competing cooperatively or greedily</td>
<td></td>
</tr>
<tr>
<td>probability</td>
<td></td>
</tr>
</tbody>
</table>
Turtle, patch, and link variables are different and each turtle (patch / link) has its own value for every other turtle (patch / link) variable.

9.2.1. List of Global Variables

**Color-adaptation**: “color-adaptation” is a switch in the user interface. When switch is on, it is possible to observe changing colors according to adaptation. In the simulation model, adaptive nodes are presented by green, resistant nodes are presented by grey, and neutral nodes are presented by red. When switch is off, all supplier nodes turn into white and then the nodes in the giant component is colored to blue.

**newcomer**: A new supplier/enterprise that is added to the market during simulation time.

**shopped?**: Shopped variable is set to true if a customer agent is on the same patch (coordinates) with a supplier agent.

**track-inventory**: “track-inventory” variable tracks total inventory in the market.

**track-money**: “track-money” variable tracks total money of costumers.

**density**: Density variable tracks the density of CMEE network.

**component-size**: Number of suppliers explored so far in the current component.

**giant-component-size**: Number of suppliers in the giant component.

**giant-start-node**: Node from where we started exploring the giant component.

**mylist**: List that holds the nodes in the giant component.

**clustering-coefficient-giant**: The clustering coefficient of the giant component. This is the average of clustering coefficients of all supplier agents in the giant component.
**average-path-length:** This variable stores the average path length of the network. In the disconnected case it stores the average path length of the giant component.

**clustering-coefficient-of-lattice:** The clustering coefficient of the initial lattice.

**average-path-length-of-lattice:** Average path length of the initial lattice.

**infinity:** A very large number. It is used to denote distance between two turtles which don't have a connected or unconnected path between them. It is set to 99999 in the simulation.

**child-suppliers:** List that holds the child suppliers which are a result of reproduce method. If a supplier’s energy reached to reproduction threshold, reproduce method fires and a child supplier joins to the ecosystem.

**supply-available?** This is a boolean (0 or 1) variable that initially set to true. If a customer demand arrives and there is enough supply in the supplier, it is set to false so that concurrency of code for supplier agents is limited.

**global-best-x:** $x$ coordinate of best fitness value found in the ecosystem by customers.

**global-best-y:** $y$ coordinate of best fitness value found in the ecosystem by customers.

**global-best-val:** Highest fitness value found in the ecosystem by customers.

**true-best-supplier:** Supplier with the best fitness-value in the ecosystem.

**particle-inertia:** Inertia of a costumer agent, which treated as a particle, in modified particle swarm optimization.

**attraction-to-personal-best:** Scalar, integer value of each agent, which represents the strength of attraction to personal best.

**attraction-to-global-best:** Scalar, integer value of each agent, which represents the strength of attraction to global best.


**particle-speed-limit:** It’s the maximum speed that customers can move in the system.

**9.2.2. List of Supplier Variables**

**explored?**: This boolean variable is used to mark suppliers we have already visited and added to giant component.

**added?**: This boolean variable is used to mark suppliers we have already added to the list, mylist.

**node-clustering-coefficient:** Clustering coefficient of each supplier.

**node-clustering-coefficient-giant:** Clustering coefficient of each supplier in the giant component.

**distance-from-other-turtles:** List of distances of this supplier agent (node) from other supplier agents.

**adapted?**: If this supplier agent variable is true, it shows that supplier agent is adapted to changes in the CMEE.

**resistant?**: If this supplier agent variable is true, it shows that supplier agent is resisting to changes in the CMEE.

**inventory:** The number of items in the inventory of a supplier agent.

**production:** The number of produced/ manufactured items by a supplier agent.

**product-price:** Product price of the products produced by a supplier agent.

**budget:** Budget of a supplier agent.

**supply:** Amount of current supplies a supplier has in the system.

**demand:** Number of current demands a supplier receives in the system.
**supplier-type:** An integer number which defines supplier type. Here, 0 corresponds to main supplier (Tier 0), 1 corresponds to Tier 1, and 2 corresponds to Tier 2.

**link-count:** Number of links a supplier agent has in CME network.

**cooperative?:** It is a boolean variable, which is true if the supplier is from a cooperative breed, and false otherwise.

**energy:** Each time a customer/enterprise buys a product from a supplier, the energy of the supplier which sells the product increases by this value.

**fitness-value:** Each supplier has a fitness value associated with it. The formula for the fitness value is as follows:

\[
\text{fitness-value} = \frac{\text{fitness-reliablity} + \text{fitness-popularity} + \text{fitness-bestprice}}{3}
\]

A customer tries to find the main supplier (Tier 0) with the best fitness. All fitness values are normalized to be between 0 and 1.

**fitness-reliability:** Each supplier has a fitness value associated with its reliability.

**fitness-popularity:** Each supplier has a fitness value associated with its popularity.

**fitness-bestprice:** Each supplier has a fitness value associated with its price.

### 9.2.3. List of Customer Variables

**money:** Money each costumer currently have and is available to spend.

**salary:** Salary each costumer agent earn in each tick.

**vx:** Velocity of the customer in the $x$ direction.

**vy:** Velocity of the customer in the $y$ direction.
**personal-best-val:** Best value customer has run across so far according to customer supplier preference.

**personal-best-gval:** Best value customer has run across so far in CMEE.

**personal-best-x:** $x$ coordinate of the best value.

**personal-best-y:** $y$ coordinate of the best value.

**customer-type:** This variable holds the customer agents supplier preferences. Different attributes of suppliers might be attractive to a customer. In our simulation customers have shopping preferences according to; reliability, popularity, and product price.

### 9.3. Process Overview and Scheduling

In the multi-agent simulation model, scheduling of the events pursuit the following order: First, environmental conditions are checked. Then customers move according to modified particle swarm optimization. Next supplier procedures are called. In each tick of simulation, supplier agents try to full-fill customer demand, buy new inventory, update their budget, supply and inventory, check their fitness values for survival, check their fitness values for opening new stores (reproduce) and propose to merge with competitors. Then the network statistics are calculated in the following order:

- Find density
- Find components of network graph
- Find giant component
- Calculate average path length
• Calculate diameter
• Calculate clustering coefficient

Next, the balance procedure is called which checks the number of each supplier type in the virtual world and balances the number of each supplier type. Then a termination condition is checked. After that, the tick counter is advanced by one. And finally, simulation output in user interface is updated.

9.4. Design Concepts

9.4.1. Emergence

Complex adaptive systems follow certain natural laws or rules, but nothing is formally planned in advance so that the behavior of a CAS is emergent. The behavior of the CMEE is said to “emerge” from the actions and interactions of its enterprises. In general, the system and its individual enterprises emerge to result in a deterministic operating environment. This emergence of highly structured collective behavior over time from the interaction of the simple entities leads to fulfillment of orders.

However, we also observe undesirable emergent phenomena when factors, such as sudden technology advancements, changing social trends, and conflicting objectives between enterprises in a CMEE arise. Those emergent phenomena may cause demand decline or amplification and therefore inventory swings. Further it may increase competition that arises in the form of sharing and contention of resources. As a
conclusion global control over agents is an exception rather than a rule; more likely it is a localized cooperation out of which a global order emerges, which is itself unpredictable.

9.4.2. Adaptation

Although it is true that individual enterprises may obey the deterministic selection process (Choi and Hartley 1996), the organization of the overall CMEE emerges through the natural process of order and spontaneity. In other words, all enterprises operate according to self-interest and to promote their own fitness criteria and thus, find order and spontaneity over a course of time. The implication of this for individual enterprises is that they need to constantly observe what emerges from a CMEE and make adjustments to organizational goals and supporting infrastructure.

Further, enterprises should realize that it is normal to behave in a deterministic fashion based on a few salient rules and performance measures. The key is to stay fit and agile and be willing to make appropriate adjustments in the face of the changing environment and not be apologetic about making structural changes over a course of time. Given the short product life of many new electronic products, manufacturing processes and technologies computer manufacturing enterprises should be designed to be adaptive to change.

Enterprises that adjust goals and infrastructure quickly, according to the changes in their customers, suppliers, and/or competitors survive longer in their CMEE than enterprises
that adhere to predetermined, static goals and infrastructure and are slow to change. To observe the adaptive behavior, we create three different states for each agent:

- “Adapted” is the state in which a supplier agent introduces new technologies or produces new products with current technologies.
- “Neutral” is the state in which a supplier agent observes the market and decides to be either one of the two states: adapted or resistant.
- “Resistant” is the state in which a supplier agent is resistant to new technological changes and not flexible enough to adapt its product line to new technological advances.

9.4.3. Fitness

Fitness typically refers to the well being of a complex aggregate of both global and local states within the system. Enterprises attempt to increase “fitness” by attending to a few simple dimensions such as delivery, cost, quality, and flexibility. For instance, a computer manufacturing enterprise tries to increase fitness by ensuring the acceptable cost, delivery, and quality purchased by building a good network of business partners. As the enterprises share their common norms and procedures, transaction costs reduce and communication efficiency increases. Some examples for fitness values from the simulation are as follows:

- fitness-reliability is equal to supply + budget + energy
- fitness-popularity is equal to demand
• fitness-bestprice is equal to inverse of product-price

• fitness-value is equal to fitness-reliability + fitness-popularity + fitness-bestprice

The enterprise survival or extinction in the enterprise ecosystem is determined by environmental fitness threshold values. For example environmental fitness threshold value for survival represents the necessary fitness level that each enterprise in the environment must sustain in order to “exist/ live” in the enterprise ecosystem.

In that sense each enterprise has a fitness value and must be fit for their tasks in order to be a viable member in the environment. For example, every enterprise must sustain some level of financial viability, if it falls below this level, the enterprise may file to bankruptcy and die in the simulation. Computer manufacturing enterprises behave in a manner so as to increase “fitness” of the system that they belong to either locally or globally. Some examples for environmental fitness threshold values from the simulation are as follows:

• If energy is less than zero, then die.

• If budget is less than zero, then die.

• If supply is less than a set value, then die.

• If inventory is less than a set value, then die.
9.4.4. Prediction

An important characteristic of CMEE is that it is a result of dynamic process. Each enterprise within the CMEE is constantly “in motion”: doing industrial / business type services, operating and transforming. CMEE emerges from this constant change.

CMEE only exists while its members are transforming, adapting and synchronizing. So that CMEE should be agile. In this respect, a complex adaptive system (CAS) perspective is well suited for modeling structural and behavioral dynamics that are present in the CMEE.

The enterprise systems as a CAS emerge from interactions among nodes in the network, which evolve over time, driven by node-level decision and rules, and environmental conditions. Therefore, the computer manufacturing enterprises are connected via links in our simulation. They have local knowledge of enterprise network via their link-neighbors which means their decisions and predictions are based on the local information and market averages.

In the real world, rapid pace of innovation in hardware and software technology creates a constant demand for newer and faster products and applications in CMEE. Being the first enterprise to market a new or better product can mean success for both the product and the enterprise. Even for many relatively commonplace items, enterprises continue to result in better, cheaper products with more desirable features.
For instance, a company that develops a new kind of computer chip that can fit into many brands of computers can earn millions of dollars in sales until a competitor is able to improve on that design. This dynamism puts a greater emphasis on prediction in CMEE than is typical in most manufacturing enterprise systems.

9.4.5. Sensing

Generally concept of sensing consists of a global view and local view. Global view requires the knowledge of state and actions of other agents. In our simulation, customer agents have the global view, so that they know brand preferences and favorites of other customer agents. This is a realistic assumption in today’s information access via Internet and social networks.

However, supplier agents have a local view. Therefore, each supplier agent has incomplete information of the state, actions, current decisions, future plans and past performance of other agents. Each supplier agent’s sensing is limited by its link neighbors. It decides its current and future actions according to its local neighbors (link neighbors) and market averages.
9.4.6. Stochasticity

Stochasticity is used to represent many sources of variability in the simulation that are too complex to represent analytically. Real enterprises make many decisions and real markets have many dynamics. In order to present this variability, the stochasticity is used in adaptive and cooperative behavior of suppliers, adding new suppliers to the system, selecting the type of new suppliers and connecting them to existing suppliers, and also balancing number of customers and suppliers in the system.

*Stochasticity in adaptive behavior:* Stochasticity in adaptive behavior can be summarized as follows: The neutral link neighbors of adaptive suppliers decide to become adaptive if a randomly created number is less than “chance-of-change”. The adaptive suppliers decide to become neutral if a randomly created number is less than a threshold “p”. Then two more random numbers are created. If the first random number is less than “lose-interest-to-change”, and the second random number is less than “resist-change”; neighbor suppliers decide to become resistant. Otherwise, if the first random number is less than “lose-interest-to-change”, and the second random number is greater than “resist-change”; neighbor suppliers decide to become neutral.

*Stochasticity in cooperative behavior:* If a randomly selected number is less than cooperative-probability, then the supplier cooperation attribute is set to greedy, otherwise it is set to cooperative.
**Stochasticity in adding new suppliers:** If a randomly selected number is greater than a threshold “p”, create a new random supplier. Note that suppliers also added to the system via reproduction.

**Stochasticity in selecting the type of new suppliers:** Create a random number less than one. If this number is less than 0.33, then set supplier type to zero (creates Tier 0 suppliers). If this number is less than 0.66, then set supplier type to one (creates Tier 1 suppliers). Otherwise, set supplier type to two (creates Tier 2 suppliers).

**Stochasticity in connecting new suppliers to existing suppliers:** Create a random number less than ten. If this number is less than 5, then randomly attach the new supplier to its closest neighbor. Otherwise, make preferential attachment.

**Stochasticity in balancing number of customers and suppliers:** If number of suppliers is greater than one fifth of the number of customers, select some customers randomly and eliminate them.

### 9.4.7. Collectives

Customer agents are categorized into three groups according to the following criteria:

- First group of customers think that reliability of suppliers is the most important.
• Second group of customers think that popularity of suppliers is the most important.

• Third group of customers think that product price of suppliers is the most important.

Supplier agents are grouped with respect to their vertical position in the supplier chain.

• Tier 0 sells to customers and buys from tier 1

• Tier 1 sells to tier 0 and buys from tier 2

• Tier 2 sells to tier 2 and its inventory increases by a constant number.

9.4.8. Observation

Observations are made via output files, graphs on interface and behavior space (behavior space is a software tool integrated with NetLogo that allows users to perform experiments with models). Further details of observations are provided in section 9.10.

9.5. Initialization

The multi-agent virtual world starts running by first initializing all global variables and then creating supplier agents. Each supplier agent’s coordinates are randomly initialized on the world. Each supplier agent is set to be “neutral” to any changes in the world. Then “initial-adaptation-size” of them is selected randomly and set to be “adapted”. Initially, each supplier agent’s number of links is set equal to zero. Their budget and inventory parameters are set to
100, supplies are set to 50, energy levels are set equal to four times that of their metabolism rates and production rates are set to a random integer between 5 and 14. Their cooperative characteristics are set randomly. Their default shapes are initialized according to their types.

Secondly customers are created. Each customer agent’s default shape is initialized to “person” and then it is randomly initialized on the world with random x and y coordinates, velocity, money, salary and type when the model starts. Personal best value is set to zero. Personal best x coordinate set to customer agent’s x coordinate and similarly, personal best y coordinate set to its y coordinate. Its color is set according to its type; pink, orange and yellow, if customer agent is type 1, type 2 and type 3, respectively.

After that we create the initial links between the suppliers and calculate the initial network statistics.

9.6. Input

The only input data that is provided to the model is the initial number of Tier 0 and Tier 1 suppliers and the controllable variables. Those variables are supplied to the model via user interface.
9.7. Submodels

The multi-agent simulation model has more than a total of 50 submodels. These submodels are categorized into four major submodels. The details of these submodels are covered in Appendix A.

9.8. Interface

The multi-agent simulation model has a detailed user interface. The components of the user interface include; buttons, sliders, switches, input variables and monitored variables. The details of the user interface components are covered in Appendix B.

9.9. Performance Measures

CMEE, which consists of supply chains with their interactions and inter-dependencies, global and local economies and customers, has candidate performance measures such as customer satisfaction, optimization and integration of information and material flows, cost minimization, profit maximization, effective risk management, response time minimization, and supplier reliability.

In this study, our aim is to observe how the number of customers, suppliers and simulation clock vary in each run. We investigate how market averages change and how the order,
inventory and supply volumes correlate with environmental conditions in the market. We study how different market behaviors of suppliers influence customers’ purchasing behavior.

Also we analyze how the market conditions affect suppliers. We observe the relation between the number of suppliers and the competitive behavior of suppliers. Moreover, we observed the adaptive trend of the market. Therefore, the performance measures that we studied in this simulation are as follows:

- Number of suppliers in the system
- Number of customers in the system
- Ecosystem life time which corresponds to CMEE survival as a whole.
- Deviation of order, inventory and supply volume
- Cooperative behavior of the market
- Adaptive behavior in the market

In the simulation, two termination conditions are present: (1) if the number of any supplier type goes to zero then the simulation ends, (2) if all suppliers are resistant to changes in the market, then the simulation ends. In order to observe the effect of the first termination condition, the second termination condition is deactivated. Then the simulation is run with the same inputs for 1000 iterations and the following variables are recorded:

- Number of Tier 0 suppliers when termination condition is met
- Number of Tier 1 suppliers when termination condition is met
- Number of Tier 2 suppliers when termination condition is met
- Number of customers when termination condition is met
- Simulation clock / ticks when termination condition is met

In addition to the observed variables listed above, the effects of environmental conditions are also recorded. It is a well-known fact that predicting the exact future behavior of a complex system, such as a CMEE, is not possible; however, this does not imply that the future is random. In fact, complex systems exhibit patterns of behavior that can be considered archetypal or prototypical (Choi, Dooley et al. 2001; Mitsuishi 2008). The existence of such a pattern in CMEE is the environmental conditions.

What we mean by environmental conditions is the economic state of the market. There is always a good time followed by a recession and sometimes even a severe economical condition, like depression. In order to model this, we added different types of environmental conditions to our ABM, such as environmental conditions representing (1) the good economic times, (2) normal economic times, and (3) severe economic times. According to the effects of those conditions, we observed the following variables:

- Customers purchase power
- Customer demand
- Amount of raw material
- Companies’ production rate
- Sales
- Product price
• Energy of companies

• Budget of companies

Moreover, we observed the effects of limited income threshold, which is a parameter that is directly proportional to the number of limited income people in the simulation model, on the number of suppliers. Furthermore, since the supplier market is treated as an ecosystem, we include the concept of reproduction in the simulation. In the CMEE, reproduction refers to branching of an enterprise which grows and reaches a certain threshold. This threshold in the simulation is referred to as the reproduction threshold. The effects of reproduction threshold on the number of suppliers are observed to measure the stability of each type of supplier (main supplier, Tier 1 and Tier 2) in the market.

In addition, the cooperative behavior of the market is observed. We recorded the numbers of cooperative agents and greedy agents for 1000 simulation runs and we observe the market whether it is adopting a certain cooperative behavior type.

Similarly, we observed the adaptive behavior trend in the market. We recorded the three different types of supplier agents: adaptive, neutral and resistant, respectively. We observed whether the market is adopting a certain adaptive behavior type and if so how this adaptive behavior type is effecting the enterprise survival.
9.10. ABM Network Analysis

From a network analysis perspective, computer manufacturing enterprises are treated as nodes. Collaboration between enterprises, information flow and supply-demand links play the role of links that are built or broken according to environmental rules and thresholds in the enterprise system. We investigated the computer manufacturing enterprise network structure by computing network properties that follow. We iteratively run the simulation 1000 times and find the range of each network statistics.

9.10.1. Network Demographics of CMEE

Network is initialized with a total of twenty suppliers: 5, 7 and 8, for Tier 0, Tier 1 and Tier 2 suppliers, respectively. Each supplier node has the following attributes in the network:

- Inventory
- Production rate
- Product price
- Budget
- Supply
- Demand
- Energy
- Fitness value associated with its reliability
- Fitness value associated with its popularity
• Fitness value associated with its product price
• Fitness value that is the total of fitness values associated with reliability, popularity and product price

Links between suppliers are created according to their types with different rules. Links between enterprises are multi-functional since they are used to transfer information, supply and demand and also represent collaborations.

9.10.2. Network Density of CMEE

The computer manufacturing enterprise network’s density represents the intensity of information flow and level of information sharing in the network relative to the total number of possible links in the network. In this sense we investigate whether the computer manufacturing enterprise network is sparse or dense.

9.10.3. Average Degree of CMEE

The average degree of this network reflects the average volume of information flow and sharing that may have influenced the CMEE at hand. The initial average degree of network is initialized to 3 so that the first time the virtual world is initialized, a supplier has 2 vertically integrated neighbors and 1 horizontally integrated neighbor.
9.10.4. Average Shortest Path Length of CMEE

We investigate the typical distance, average shortest path length, between a pair of enterprises in terms of their multi-functional links. The average shortest path length of a connected network is the average of the shortest paths that exist between all possible pairs of nodes in a network. The formula for average shortest path length is described in section 4.2.5 of chapter 4.

In NetLogo, the “Floyd Warshall algorithm for all pair’s shortest paths” (Wilensky 2005) is implemented for path length computations. It is a dynamic programming algorithm which builds bigger solutions from the solutions of smaller subproblems using memorization that is storing the results. It keeps finding incrementally if there is shorter path through the network. The algorithm iterates over all supplier agents, and finds the shortest possible path for each supplier agent.

The average path length of computer manufacturing enterprise network shows on average how many intermediary enterprises are required for an enterprise in the business environment to operate with any other enterprise in the network.

9.10.5. Clustering Coefficient of CMEE

Watts and Strogatz (Watts and Strogatz 1998) introduced an important concept of clustering in social networks. The clustering coefficient characterizes the local transitivity and order in the neighborhood of a node. The clustering coefficient of node $i$ is the ratio
of the number of edges $E_i$ that actually exist among the one-link-away neighborhood nodes to the maximum number of possible edges among these neighboring nodes.

In the present study, the clustering coefficient for the computer manufacturing enterprise network refers to the probability that any two collaborator enterprises of a third enterprise have collaborated themselves.

### 9.10.6. Small World Network Characteristics of CMEE

A short path length and high clustering coefficient indicates small world characteristics. The existence of the small-world effect was first demonstrated by the famous experiment conducted by Stanley Milgram in the 1960s (Milgram 1967). It led to the popular concept of six degrees of separation. Milgram found that there is an acquaintance path of an average length of 6 in the social network of people in the US. Using the same analogy, we investigate whether small world characteristic exists in the computer manufacturing enterprise network.

### 9.10.7. Percolation Theory of CMEE

One of the most interesting properties of graph theory is percolation transition. According to this theory, there exists a critical probability $p_c$ below which the network is a collection of isolated clusters and above which the network forms a giant component quite rapidly (Albert and Barabasi 2002). By using analogy of percolation transition, we analyze
connectivity of the computer manufacturing enterprise network starting with its first largest connected component.

9.10.8. Diameter of CMEE

The diameter of a network is the longest shortest path between two vertices. So one considers every pair of vertices, finds the shortest path between each of the pairs, and then determines the diameter as the longest of these shortest paths. In our study, the diameter represents the maximum separation distance among pairs of enterprises. This parameter indicates the greatest distance an enterprise will ever have to hop to reach another enterprise in the computer manufacturing enterprise network. In other words, for the computer manufacturing enterprise network, the diameter refers to the length of the chain of information flow connecting any two enterprises is less than or equal to a constant, namely diameter.
Chapter 10

Testing and Results

The results, testing, validation and verification of the agent-based simulation analysis are presented in this chapter. For testing, verification and validation of the model, we use the well-established methods (Pritsker and Pegden 1979; Kelton, Sadowski et al. 1998; Law 2007; Bruegge and Dutoit 2009).

In the model verification step, we check if the accuracies of the model are consistent with their intended objectives. In order to verify the model; the simulation model is run 1000 times and customer entities are sent to the system. The customer entities and their chain of interactions are observed in the system and the evaluated results are traced for those entities. In addition to the trace results, additional features are used to control the model execution and step through the system.

The animation is used to verify the system. Each resource and group are animated and examined. By examining the animation of the model, we verify whether each enterprise follows the right sequence of evolution steps through the CMEE. The model is also validated through feedback provided by thesis advisers who have deep insights into the dynamics of real-world CMEE.
10.1. Testing

Software testing is the process of validating and verifying that a software program, application or product meets the requirements that guided its design and development, works as expected, and can be implemented with the same characteristics. Here, the unit testing, integration testing, system testing, white box and black box testing techniques are applied to validate and verify the agent based simulation of CMEE.

10.1.1. Unit Testing, Integration Testing and System Testing

In computer programming unit testing is a method by which individual units of source code are tested to determine if they are fit for use. A unit is the smallest testable part of an application. NetLogo is a procedural language. In NetLogo, each individual submodel corresponds to a unit.

After unit testing, an integration testing technique was applied to the source code. Integration testing is the phase in software testing in which individual software modules are combined and tested as a group. In integration testing, modules, which have been unit tested before, are grouped in larger aggregates and tested. Finally, we applied system testing to whole source code.

In our model, we tested each submodel separately in smaller simpler models before we integrated them into a single software system. We re-tested each submodel after
integrating it with the whole system. Some of the important individually tested algorithms are as follows:

- Preferential attachment
- Calculating average path length
- Calculating clustering coefficient
- Finding giant component of network
- Adaption of suppliers agents
- Agility of suppliers agents
- Alignment of suppliers agents
- Modified particle swarm optimization for customer agents

Some of the important integration tested algorithms are as follows:

- Network statistics:
  - Calculating average path length
  - Calculating clustering coefficient
  - Calculating density
- Supplier behavior:
  - Adaption of suppliers agents
  - Agility of suppliers agents
  - Alignment of suppliers agents
10.1.2. Black Box Testing

Black-box testing is a method of software testing that tests the functionality of an application as opposed to its internal structures or workings. Test cases are built around specifications and requirements, i.e., what the application is supposed to do. It uses external descriptions of the software, including specifications, requirements, and designs to derive test cases.

*Example case:* Termination conditions for supplier agents.

Valid and invalid inputs are tested for supplier agents such as:

- A supplier agent’s “who number” (who number is a built-in turtle variable, which holds the turtle's ID number) is recorded and budget attribute is set to -5
- Then we check if this supplier is still alive or not
- Correct output is “supplier is dead”.

The snippet code for this example case is:

```lisp
to-report test-code-part-1
  let mywho 0
  let me nobody

  ask one-of suppliers [  
    set budget -5  
    set mywho who  
    set me myself
  ]

  report me
end
```
to test-code-part-2 [ me ]
  ifelse me != nobody
    [ print "error: supplier"
      type [who] of me
      type "is still alive"
    ]
    [ print "code works fine"
  end

10.1.3. White Box Testing

White-box testing is a method of testing software that tests internal structures or workings of an application, as opposed to its functionality. In white-box testing an internal perspective of the system, as well as programming skills, are employed to design test cases.

Example case “Branch Testing”: Inputs were chosen to exercise paths through the following piece of code and appropriate outputs were determined.

- Choose a supplier with “resistant-size + adaptive-size” greater than “0.8 * n”, and then check if it is neutral to changes in the market, which means that both attributes “adapted?” and “resistant?” are equal to false.

- Choose a supplier with “adaptive-size” less than “0.2 * n”, and then check if it is adaptive to changes in the market, in other words, attribute “adapted?” is equal to true and “resistant?” is equal to false.

- Choose a supplier with “resistant-size” less than “0.2 * n”, and then check if it is resistant to changes in the market, i.e., attribute “adapted?” is equal to false and “resistant?” is equal to true.
• Choose a supplier with “resistant-size” less than “adaptive-size”, and then check if it is adaptive to changes in the market, i.e., attribute “adapted?” is equal to true and “resistant?” is equal to false.

• Choose a supplier with “adaptive-size” less than “resistant-size”, and then check if it is resistant to changes in the market, i.e., attribute “adapted?” is equal to false and “resistant?” is equal to true.

The snippet code for this example case is:

```plaintext
ifelse resistant-size + adaptive-size > 0.8 * n
    [ realize-change-in-market ]
    [ ifelse adaptive-size < 0.2 * n
        [ become-adapted ]
        [ ifelse resistant-size < 0.2 * n
            [ become-resistant ]
            [ ifelse resistant-size < adaptive-size
                [ become-adapted ]
                [ become-resistant ]
            ]
        ]
    ]
```

10.2. Results of Statistical Analysis

After verification and validation of the agent based simulation model of CMEE, we run the model to build and analyze the triple-A virtual CMEE. We performed 1000 simulation runs and recorded supplier and customer agent attributes and environmental parameters in the system.
As expected, the deviation of the number of different customer types in the system is close to each other, mostly following the environmental condition trends. However, the deviation of number of different supplier types in the system follows a more interesting trend. It is observed that 64 percent of the time termination condition 1 (if the number of any of the supplier type goes to zero then end simulation) is met because Tier 2 suppliers are extinct. However only 1 percent of the time termination condition 1 is met because of the main suppliers, which represent the companies at the top of the vertical integration are stronger in the market and survive longer. Graphs of deviations of different type of customers and deviations of different type of suppliers at the end of each simulation run are shown in Figures 10.1 and 10.2.

Also, the deviation in simulation termination time in the system is observed. The statistical results show that there is a significant deviation in market life expectancy according to current environmental conditions of the market. In the simulation model three different types of environmental conditions, the good economic times, normal economic times, and severe economic times produce the results stated below:

- Customers’ purchasing power decreases when environmental conditions get harsher; customers with a limited budget are unwilling to spend money and hence the demand for products decreases
- As a result of decreasing customer demand sales decreases
- Because of decreasing sales, companies decrease the product price to vitalize product sales
Figure 10.1 Number of customer agents versus simulation clock

Figure 10.2 Number of supplier agents versus simulation clock
Chapter 10

Figure 10.3 Market attributes versus environmental conditions

- Decreasing sales compels companies to decrease the production rate, therefore they order fewer raw material and keep fewer inventory
- Since both sales and production rate decrease, energy and budget of companies decrease significantly

One can see that one negative effect in the market creates a domino effect and affects every layer of the supply chain. All these results can be observed in Figure 10.3.

In Figure 10.4, the supplier behavior with respect to changing reproduction threshold is observed for the three different environmental conditions. The reproduction threshold is the
measure of difficulty level beyond which a supplier is able to grow enough and branch out. The general observation from this graph is the inverse relationship of decreasing number of suppliers with respect to increasing threshold values.

The first graph of Figure 10.4 shows “reproduction threshold versus number of suppliers” in good economic conditions. In this graph, first, the number of main suppliers decrease with increasing reproduction threshold then it makes a significant increase for a while, and then it decreases again as a result of very high threshold values. In order to explain this sudden increase, one has to consider the behavior of Tier 1 and Tier 2 suppliers when the reproduction threshold is increased; we have to also keep in mind that the market conditions are at their best values. With the effect of increasing threshold, the first reaction observed in the market is the decreasing number of main and Tier 1 suppliers. However, small number of Tier 2 suppliers and decrease in number of Tier 1 suppliers provide a competitive advantage to main suppliers and they start to grow in numbers until the threshold reaches a certain value. The Tier 2 suppliers have a more stable profile compared to the other two suppliers. They are smaller in number and not affected by the changes in reproduction threshold.

In second graph, a similar behavior to the first graph is observed but the amount of growth is more limited since now market conditions have changed from good to normal. Finally, under severe market conditions, the market does not create any opportunities for competitive advantage and so the observed behavior is downward in the third graph of Figure 10.4.
Figure 10.4 Number of suppliers versus reproduction threshold for different environmental conditions
Next, we conducted a design of experiments study, to determine the effect of two factors: reproduction threshold and environmental conditions on the market survival. The factors and factor levels are shown in Table 10.1.

Table 10.1 The factors and factor levels

<table>
<thead>
<tr>
<th>Factors</th>
<th>Label</th>
<th>Low Level</th>
<th>High Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Reproduction threshold</td>
<td>205 (low)</td>
<td>225 (high)</td>
</tr>
<tr>
<td>2</td>
<td>Environmental conditions</td>
<td>1 (good)</td>
<td>3 (harsh)</td>
</tr>
</tbody>
</table>

Response values (market survival) for each combination were obtained by running the simulation model using NetLogo. The factors, factor levels and response values were input MATLAB functions to conduct DOE analysis. The results are shown in Table 10.2 and Table 10.3.

Table 10.2 The standard ANOVA table of adaptation size of adaptive suppliers

<table>
<thead>
<tr>
<th>Source</th>
<th>SS</th>
<th>df</th>
<th>MS</th>
<th>F</th>
<th>Prob &gt; F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Columns</td>
<td>2832825.6</td>
<td>1</td>
<td>2832825.6</td>
<td>128.03</td>
<td>0</td>
</tr>
<tr>
<td>Rows</td>
<td>199005.2</td>
<td>1</td>
<td>199005.2</td>
<td>8.99</td>
<td>0.0029</td>
</tr>
<tr>
<td>Interaction</td>
<td>200345.8</td>
<td>1</td>
<td>200345.8</td>
<td>9.05</td>
<td>0.0028</td>
</tr>
<tr>
<td>Error</td>
<td>8762122.9</td>
<td>396</td>
<td>22126.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>11994299.4</td>
<td>399</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 10.3 Results for design of experiments study

<table>
<thead>
<tr>
<th>Coefficient</th>
<th>Value of the Coefficient</th>
<th>Null Hypothesis</th>
<th>p value</th>
<th>T-Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\beta_0$</td>
<td>904.8492</td>
<td>$H_0: \beta_0 = 0$</td>
<td>0.0000</td>
<td>Reject $H_0$ ($p&lt;0.01$)</td>
</tr>
<tr>
<td>$\beta_{rep}$</td>
<td>-2.6848</td>
<td>$H_0: \beta_{rep} = 0$</td>
<td>0.000</td>
<td>Reject $H_0$ ($p&lt;0.01$)</td>
</tr>
<tr>
<td>$\beta_{env}$</td>
<td>-290.5122</td>
<td>$H_0: \beta_{env} = 0$</td>
<td>0.0029</td>
<td>Reject $H_0$ ($p&lt;0.01$)</td>
</tr>
<tr>
<td>$\beta_{rep&amp;env}$</td>
<td>0.8974</td>
<td>$H_0: \beta_{rep&amp;env} = 0$</td>
<td>0.0028</td>
<td>Reject $H_0$ ($p&gt;0.01$)</td>
</tr>
</tbody>
</table>

We used analysis of variance (ANOVA) to identify which of the model terms are likely to have no effect on the response (i.e., for which terms $\beta_i$ and $\beta_{ij}$ are likely to be zero) at the 1% significance level. Based on the $p$ values from Table 10.2, we see that factors reproduction threshold and environmental conditions are statistically significant at the greater than 99.71% level and their interaction is also significant at the 99.72% level.

According to above results in Table 10.3, the main effect of Factor 1 (reproduction threshold) is significant ($p = 0.000 < 0.01$) and has a significant effect of -2.6848. This means that the average effect of increasing reproduction threshold in the CMEE decreases the length of market survival by 2.6848. The main effect of Factor 2 (environmental conditions) is significant ($p = 0.0029 < 0.01$) and has a significant effect of -290.5122. This means that the average effect of changing environmental conditions from good to harsh in the CMEE
decreases the length of market survival by 290.5122. The interactive effect of these two factors is significant \((p = 0.0028 > 0.01)\) and has a significant effect of 0.8974. This means that setting both factor at low level (low reproduction threshold & good environmental conditions) increases the length of market survival by 0.8974 in CMEE.

Since all the factors are statistically significant, we can predict the effect of the two factors – reproduction threshold and environmental conditions – on market survival by using their main effects and the interaction between factors. We can also write down the regression equation of the system as follows:

\[
E[R\text{(reproduction threshold, environmental conditions)}] = \beta_0 + \beta_{rep} \cdot n_{rep} + \beta_{env} \cdot n_{env} + \beta_{rep\&env} \cdot n_{rep\&env}
\]  
(10.2.1)

\[
E[R\text{(reproduction threshold, environmental conditions)}] = 904.8492 - 2.6848 \cdot n_{rep} - 290.5122 \cdot n_{env} + 0.8974 \cdot n_{rep\&env}
\]  
(10.2.2)

Above statistical results show that there is a significant decrease in market life expectancy of the CMEE when current environmental condition changes from good to harsh; reproduction threshold (decreasing enterprise ability of growing and branching out) increases from low to high and combination of these two factors at harsh environmental conditions and high reproduction threshold values.
Figure 10.5 addresses the question of how the numbers of adaptive, neutral and resistant suppliers change in time for one simulation run. In order to see whether the CME is adopting a certain adaptive behavior type and if so how this adaptive behavior type is affecting the enterprise survival, we applied the statistical methods: such as confidence interval on mean and one way ANOVA of the number of adaptive, neutral and resistant suppliers. In general, we name the number of adaptive, neutral and resistant suppliers as adaptation size. Separately, each variable named as adaptation size, neutral size and resistant size for the number of adaptive, neutral and resistant suppliers, respectively.

In the beginning of the study, we have decided that the initial number of total suppliers in the network is 20 and the initial number of adaptive suppliers is 10. Therefore, the system design alternatives with “the initial number of adaptive suppliers of 4, 5, 6, 7, 8, 9” based on the “average number of adaptive, neutral and resistant suppliers” performance measures were evaluated to observe adaptivity trend of CME.

Figure 10.6 gives the 99% confidence interval for the “average number of adaptive, neutral and resistant suppliers for different initial number of adaptive suppliers” performance measure. It can be clearly seen that the number of adaptive suppliers at the end of each simulation run is more than the number of neutral and resistant suppliers in the market. It proves that enterprises, which adjust goals and infrastructure quickly according to the changes in the customers, suppliers, and/or competitors, survive longer in the CME. However, results also point to an interesting result. The resistant suppliers are more successful in survival than neutral suppliers.
Figure 10.5 Adaptivity trends of neutral, adaptive and resistant suppliers

Figure 10.6 C.I. on mean adaptation size for neutral, adaptive and resistant suppliers
This shows the fact that transition period of adapting new technologies in the market is a risky period for an enterprise. During the transition period, as a result of conflicting goals and difficulties in changing current infrastructure, enterprises may become vulnerable and fail to survive in the CMEE.

Next, we performed balanced one-way ANOVA for comparing the means of “average number of adaptive, neutral and resistant suppliers for different initial number of adaptive suppliers” performance measure. Here, each different initial number of adaptive suppliers represents an independent sample containing mutually independent observations. The $p$ value for the null hypothesis is calculated under the assumption that all samples are drawn from populations with the same mean. Therefore, if $p$ is near zero, it casts doubt on the null hypothesis and suggests that at least one sample mean is significantly different from the other sample means. The common significance level is set to 0.01.

The standard ANOVA table and the box plot of these comparisons for adaptive, neutral and resistant suppliers are provided in Table 10.4 and Figure 10.7, Table 10.5 and Figure 10.8, and Table 10.6 and Figure 10.9, respectively.

The standard ANOVA table divides the variability of the data into two parts:

- Variability due to the differences among the means of each group of data (variability between groups)
Variability due to the differences between each group of data and the mean of each group of data (variability within groups)

The standard ANOVA table has six columns:

- The source of the variability.
- The sum of squares (SS) due to each source.
- The degrees of freedom (df) associated with each source.
- The mean squares (MS) for each source, which is the ratio \( \frac{SS}{df} \).
- The \( F \)-statistic, which is the ratio of the mean squares.
- The \( p \) value, which is derived from the cumulative distribution function of \( F \).

The box plot suggests the size of the \( F \)-statistic and the \( p \) value. Large differences in the center lines of the boxes correspond to large values of \( F \) and correspondingly small values of \( p \).

From the ANOVA table output of adaptive and resistant suppliers, one can see that the confidence intervals for multiple comparisons all lie above zero. Therefore, the null hypothesis is accepted, there is no difference between the means of adaptive and resistant suppliers. However, ANOVA table output of neutral suppliers suggests that there is significant evidence that the means of neutral suppliers are not equivalent. Therefore, we performed a multiple comparison test to determine where those differences lie.
Table 10.4 The standard ANOVA table of adaptation size of adaptive suppliers

<table>
<thead>
<tr>
<th>Source</th>
<th>SS</th>
<th>df</th>
<th>MS</th>
<th>F</th>
<th>Prob &gt; F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Columns</td>
<td>1274.3</td>
<td>6</td>
<td>212.38</td>
<td>2.25</td>
<td>0.0359</td>
</tr>
<tr>
<td>Error</td>
<td>660058</td>
<td>6993</td>
<td>94.388</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>661332.3</td>
<td>6999</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 10.5 The standard ANOVA table of adaptation size of neutral suppliers

<table>
<thead>
<tr>
<th>Source</th>
<th>SS</th>
<th>df</th>
<th>MS</th>
<th>F</th>
<th>Prob &gt; F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Columns</td>
<td>838.2</td>
<td>6</td>
<td>139.701</td>
<td>25.15</td>
<td>1.05916e-029</td>
</tr>
<tr>
<td>Error</td>
<td>38839.6</td>
<td>6993</td>
<td>5.554</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>39667.8</td>
<td>6999</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 10.6 The standard ANOVA table of adaptation size of resistant suppliers

<table>
<thead>
<tr>
<th>Source</th>
<th>SS</th>
<th>df</th>
<th>MS</th>
<th>F</th>
<th>Prob &gt; F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Columns</td>
<td>426.1</td>
<td>6</td>
<td>71.0159</td>
<td>1.23</td>
<td>0.2875</td>
</tr>
<tr>
<td>Error</td>
<td>403868.2</td>
<td>6993</td>
<td>57.7532</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>404294.3</td>
<td>6999</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 10.7 One way ANOVA of adaptation size of adaptive suppliers
Figure 10.8 One way ANOVA of neutral size of neutral suppliers

Figure 10.9 One way ANOVA of resistance size of resistant suppliers
According to *multiple comparison method* results given in Table 10.7 and Figure 10.10, confidence intervals for all comparisons of means for adaptive agents contain zero. This means that there is no statistically significant difference between design of initial adaptation size of 4, 5, 6, 7, 8, 9 and 10 for adaptive agents.

Similarly, according to *multiple comparison method* results given in Table 10.8 and Figure 10.11, confidence intervals for all comparisons of means for resistant agents contain zero. This means that there is no statistically significant difference between design of initial adaptation size of 4, 5, 6, 7, 8, 9 and 10 for resistant agents. In other words, there is no statistically significant change in average number of resistant agents, when the initial number of adaptive agents is increased from 4 to 10.

Furthermore, according to *multiple comparison method* results given in Table 10.9 and Figure 10.12, the confidence intervals of neutral agents for MU(4) – MU(5), MU(4) – MU(6), MU(4) – MU(7), MU(4) – MU(8), MU(4) – MU(9), MU(4) – MU(10), lie completely above zero (Here, MU represents the mean of the null hypothesis). So, we can say that designs 5 (initial adaptation size of 5), 6 (initial adaptation size of 6), 7 (initial adaptation size of 7), 8 (initial adaptation size of 8), 9 (initial adaptation size of 9), and 10 (initial adaptation size of 10) are significantly different from design 4 (initial adaptation size of 4); because the average number of neutral agents are smaller.

The confidence intervals for MU(5) – MU(6), MU(5) – MU(7), MU(5) – MU(8), MU(5) – MU(9), MU(5) – MU(10), also lie completely above zero. This means that designs 6 (initial
adaptation size of 6), 7 (initial adaptation size of 7), 8 (initial adaptation size of 8), 9 (initial adaptation size of 9), and 10 (initial adaptation size of 10) are significantly different from design 5 (initial adaptation size of 5).

Similarly, the confidence intervals for \( \text{MU}(6) - \text{MU}(9) \), \( \text{MU}(6) - \text{MU}(10) \), \( \text{MU}(7) - \text{MU}(9) \), \( \text{MU}(7) - \text{MU}(10) \), \( \text{MU}(8) - \text{MU}(9) \), \( \text{MU}(8) - \text{MU}(10) \), also lie completely above zero. This means that designs 9 (initial adaptation size of 9) and 10 (initial adaptation size of 10) are significantly different from design 6 (initial adaptation size of 6). Also, design 10 (initial adaptation size of 10) is significantly different from all other designs except design 9.

However, confidence intervals of neutral agents for \( \text{MU}(6) - \text{MU}(7) \), \( \text{MU}(6) - \text{MU}(8) \), \( \text{MU}(7) - \text{MU}(8) \), \( \text{MU}(8) - \text{MU}(9) \), \( \text{MU}(9) - \text{MU}(10) \) contain zero. It means that there is no statistically significant difference between those designs. In other words, there is no statistically significant change in average number of neutral agents, when the initial number of adaptive agents is increased from 6 to 7, 6 to 8, 7 to 8, 8 to 9 and 9 to 10.

We can conclude from these results that the simulation model produces no statistically significant difference in the means of adaptive and resistant suppliers. However, it produces statistically significant difference in the means of neutral suppliers. Therefore, we can comment on the adaptive trend of adaptive and resistant suppliers statistically correctly. Though, the simulation model does not provide clear results for the adaptive trend of neutral suppliers.
Table 10.7 Multiple pairwise comparisons of adaptive agents performed at the 0.01 level

<table>
<thead>
<tr>
<th>Sample 1</th>
<th>Sample 2</th>
<th>Lower Limit</th>
<th>Upper limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>5</td>
<td>-0.7279</td>
<td>0.5531</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>-0.8647</td>
<td>0.4163</td>
</tr>
<tr>
<td>4</td>
<td>7</td>
<td>-1.5918</td>
<td>-0.3108</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>-1.7756</td>
<td>-0.4946</td>
</tr>
<tr>
<td>4</td>
<td>9</td>
<td>-1.5638</td>
<td>-0.2828</td>
</tr>
<tr>
<td>4</td>
<td>10</td>
<td>-1.9633</td>
<td>-0.6823</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>-1.4178</td>
<td>-0.1368</td>
</tr>
<tr>
<td>5</td>
<td>7</td>
<td>-2.1449</td>
<td>-0.8639</td>
</tr>
<tr>
<td>5</td>
<td>8</td>
<td>-2.3287</td>
<td>-1.0477</td>
</tr>
<tr>
<td>5</td>
<td>9</td>
<td>-2.1169</td>
<td>-0.8359</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>-2.5164</td>
<td>-1.2354</td>
</tr>
<tr>
<td>6</td>
<td>7</td>
<td>-2.0081</td>
<td>-0.7271</td>
</tr>
<tr>
<td>6</td>
<td>8</td>
<td>-2.1919</td>
<td>-0.9109</td>
</tr>
<tr>
<td>6</td>
<td>9</td>
<td>-1.9801</td>
<td>-0.6991</td>
</tr>
<tr>
<td>6</td>
<td>10</td>
<td>-2.3796</td>
<td>-1.0986</td>
</tr>
<tr>
<td>7</td>
<td>8</td>
<td>-1.4648</td>
<td>-0.1838</td>
</tr>
<tr>
<td>7</td>
<td>9</td>
<td>-1.253</td>
<td>0.028</td>
</tr>
<tr>
<td>7</td>
<td>10</td>
<td>-1.6525</td>
<td>-0.3715</td>
</tr>
<tr>
<td>8</td>
<td>9</td>
<td>-1.0692</td>
<td>0.2118</td>
</tr>
<tr>
<td>8</td>
<td>10</td>
<td>-1.4687</td>
<td>-0.1877</td>
</tr>
<tr>
<td>9</td>
<td>10</td>
<td>-1.6805</td>
<td>-0.3995</td>
</tr>
</tbody>
</table>
Table 10.8 Multiple pairwise comparisons of resistant agents performed at the 0.01 level

<table>
<thead>
<tr>
<th>Sample 1</th>
<th>Sample 2</th>
<th>Lower Limit</th>
<th>Upper limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>5</td>
<td>-1.4848</td>
<td>-0.4828</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>-1.7153</td>
<td>-0.7133</td>
</tr>
<tr>
<td>4</td>
<td>7</td>
<td>-1.1329</td>
<td>-0.1308</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>-1.1245</td>
<td>-0.1224</td>
</tr>
<tr>
<td>4</td>
<td>9</td>
<td>-1.5655</td>
<td>-0.5634</td>
</tr>
<tr>
<td>4</td>
<td>10</td>
<td>-1.2301</td>
<td>-0.228</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>-1.2325</td>
<td>-0.2305</td>
</tr>
<tr>
<td>5</td>
<td>7</td>
<td>-0.6501</td>
<td>0.352</td>
</tr>
<tr>
<td>5</td>
<td>8</td>
<td>-0.6417</td>
<td>0.3604</td>
</tr>
<tr>
<td>5</td>
<td>9</td>
<td>-1.0826</td>
<td>-0.0806</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>-0.7473</td>
<td>0.2548</td>
</tr>
<tr>
<td>6</td>
<td>7</td>
<td>-0.4196</td>
<td>0.5824</td>
</tr>
<tr>
<td>6</td>
<td>8</td>
<td>-0.4112</td>
<td>0.5908</td>
</tr>
<tr>
<td>6</td>
<td>9</td>
<td>-0.8522</td>
<td>0.1498</td>
</tr>
<tr>
<td>6</td>
<td>10</td>
<td>-0.5168</td>
<td>0.4852</td>
</tr>
<tr>
<td>7</td>
<td>8</td>
<td>-0.9936</td>
<td>0.0084</td>
</tr>
<tr>
<td>7</td>
<td>9</td>
<td>-1.4346</td>
<td>-0.4326</td>
</tr>
<tr>
<td>7</td>
<td>10</td>
<td>-1.0992</td>
<td>-0.0972</td>
</tr>
<tr>
<td>8</td>
<td>9</td>
<td>-1.443</td>
<td>-0.441</td>
</tr>
<tr>
<td>8</td>
<td>10</td>
<td>-1.1076</td>
<td>-0.1056</td>
</tr>
<tr>
<td>9</td>
<td>10</td>
<td>-0.6666</td>
<td>0.3354</td>
</tr>
</tbody>
</table>
Table 10.9 Multiple pairwise comparisons of neutral agents performed at the 0.01 level

<table>
<thead>
<tr>
<th>Sample 1</th>
<th>Sample 2</th>
<th>Lower Limit</th>
<th>Upper limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>5</td>
<td>-0.1838</td>
<td>0.127</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>0.1362</td>
<td>0.4469</td>
</tr>
<tr>
<td>4</td>
<td>7</td>
<td>0.2104</td>
<td>0.5212</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>0.3614</td>
<td>0.6722</td>
</tr>
<tr>
<td>4</td>
<td>9</td>
<td>0.5847</td>
<td>0.8954</td>
</tr>
<tr>
<td>4</td>
<td>10</td>
<td>0.7033</td>
<td>1.0141</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>0.0092</td>
<td>0.32</td>
</tr>
<tr>
<td>5</td>
<td>7</td>
<td>0.0835</td>
<td>0.3942</td>
</tr>
<tr>
<td>5</td>
<td>8</td>
<td>0.2345</td>
<td>0.5452</td>
</tr>
<tr>
<td>5</td>
<td>9</td>
<td>0.4577</td>
<td>0.7685</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>0.5764</td>
<td>0.8871</td>
</tr>
<tr>
<td>6</td>
<td>7</td>
<td>-0.2365</td>
<td>0.0743</td>
</tr>
<tr>
<td>6</td>
<td>8</td>
<td>-0.0855</td>
<td>0.2252</td>
</tr>
<tr>
<td>6</td>
<td>9</td>
<td>0.1377</td>
<td>0.4485</td>
</tr>
<tr>
<td>6</td>
<td>10</td>
<td>0.2564</td>
<td>0.5671</td>
</tr>
<tr>
<td>7</td>
<td>8</td>
<td>-0.1598</td>
<td>0.151</td>
</tr>
<tr>
<td>7</td>
<td>9</td>
<td>0.0635</td>
<td>0.3742</td>
</tr>
<tr>
<td>7</td>
<td>10</td>
<td>0.1821</td>
<td>0.4929</td>
</tr>
<tr>
<td>8</td>
<td>9</td>
<td>-0.0875</td>
<td>0.2232</td>
</tr>
<tr>
<td>8</td>
<td>10</td>
<td>0.0312</td>
<td>0.3419</td>
</tr>
<tr>
<td>9</td>
<td>10</td>
<td>-0.1921</td>
<td>0.1186</td>
</tr>
</tbody>
</table>

Figure 10.10 Multiple pairwise comparisons of adaptive agents
Figure 10.11 Multiple pairwise comparisons of resistant agents

Figure 10.12 Multiple pairwise comparisons of neutral agents
In the model, two kinds of competition methods are implemented: being greedy and being cooperative. These two different strategies are examined and the results are presented in Figures 10.13 and 10.14. The number of cooperative and greedy suppliers is named as cooperation size in Figure 10.14.

Figure 10.13 addresses the question of how the numbers of cooperative and greedy suppliers change while competing against each other within CMEE that evolves over time in one simulation run. According to this run, the cooperative suppliers are more successful in survival than the greedy suppliers in CMEE. In order to obtain statistically accurate results, we took a sample of 1000 runs, applied the statistical analysis of confidence interval on mean and investigated whether the CMEE is adopting this cooperative behavior in general.

Figure 10.14 gives the 99% confidence interval for the “average number of cooperative and greedy suppliers” performance measure. It can be clearly seen that the number of cooperative suppliers at the end of each simulation run is more than the number of greedy suppliers in the market.

This result proves that suppliers, when they adopt the cooperative behavior, benefit from increased collaboration and information flow with their neighbors and gain advantage over their greedy competitors. Their fitness levels increase and as a result they survive longer in the CMEE.
Figure 10.13 The number of cooperative versus greedy suppliers

Figure 10.14 C.I. on mean cooperation size
10.3. Results of ABM Network Analysis

We have calculated the network statistics using analogy of percolation transition. First, we analyze the connectivity of the computer manufacturing enterprise network. In a disconnected network, the average path length does not make sense, or perhaps may be considered infinite. Therefore, in the disconnected case, network statistics are calculated using the first largest connected component.

10.3.1. Network Density

We investigate whether the computer manufacturing enterprise network is sparse or dense. The density of the network (in 1000 runs, a total of 48000 ticks, and an average of 48 ticks per simulation) has a mean of 0.1264 with a standard deviation of 0.1280, which shows CMEE has a sparse network structure.

10.3.2. Average Degree

The average degree of network is initialized to three in the beginning of each simulation run so that a supplier can start its ecosystem life with two vertically integrated neighbors and one horizontally integrated neighbor. At the end of 1000 simulation runs, the maximum, minimum, mean and standard deviation of average degree of computer manufacturing enterprise network is found to be 5.5383, 1.7579, 3.4099 and 0.9561, respectively. These results show that enterprises in CMEE have similar number of neighbors / collaborators. Enterprises, which have many more neighbors / connections
than other enterprises, are not observed in the computer manufacturing enterprise network.

10.3.3. Average Shortest Path Length

The mean and maximum number in all runs for average path length of computer manufacturing enterprise network is found to be 2.2825 and 4.9345, respectively. This implies that any enterprise in the CMEE can reach any other enterprise in the ecosystem through a relatively small number of intermediary collaborator enterprises.

10.3.4. Clustering Coefficient

In our simulation, the clustering coefficient value $C$ indicates that a pair of enterprises has a $C$ percent chance of interacting with each other if they both have a business with a common third enterprise. It is found that the clustering coefficient value of the giant component of computer manufacturing enterprise network deviates during the life span of the market. The main interesting observation is that with increasing clustering coefficient values the market starts to grow and with decreasing values it shrinks.

10.3.5. Small World Network Characteristics

As stated in chapter 4, network theory, social networks and information networks exhibit small-world network characteristics. Similar to these results observed in literature, we
expect to observe short path length and high clustering in the computer manufacturing enterprise network which indicates small world characteristics. The observed results show that computer manufacturing enterprise network has a short path length. However, its clustering coefficient value changes during the life span of the market. When the market is growing the clustering coefficient value increases up to 0.9. It indicates that CMEE behaves as a small world, while market is active: it grows with a relatively small number of intermediary enterprises in the network (observed short path length) and has a high clustering coefficient.

10.3.6. Diameter

We also have calculated the maximum separation distance among pairs of enterprises. This parameter indicates the greatest distance an enterprise will ever have to hop to reach another node in the computer manufacturing enterprise network. This path length is referred to as the diameter of the network. For the computer manufacturing enterprise network, diameter means that the length of the chain of information flow connecting any two enterprises has a mean 4.3883 with a standard deviation of 1.4091.

The results of ABM network analysis show that on average the number of collaborators a computer manufacturing enterprise has is limited. In order to be ready to sudden changes (loss due to fire, loss due to earthquake, bankruptcy, etc. of an upstream supplier) and be more agile, computer manufacturing enterprises should increase the number of their collaborators in the CMEE. Further, one should keep in mind that increased number of
connections and collaborators in the supply chain facilitates better information flow and improves the ability to follow and adapt new technologies. A computer manufacturing enterprise should be able to better utilize its network and should be aware of the fact that in reality every other enterprise in CMEE is relatively close in terms of collaboration and information links.
Chapter 11

Conclusion and Future Work

In this dissertation, we have tried to derive and investigate adaptation mechanisms in CMEE to better understand their possible evolutionary paths. Specifically, we tried to deliver the following: (1) using agent based and network theory based methods to study evolution and self-organization of CMEE, (2) developing an accurate comprehensive model for understanding CMEE by developing an agent based simulation by considering adaptation, alignment and agility of the CMEE, (3) conducting a comprehensive study on the effect of uncertain and rapidly changing business environments in the CMEE, (4) investigating the network structure of the model simultaneously with the agent based simulation, and (5) performing statistical and network analysis of the resultant model.

In previous chapters we explained briefly the nature of computer manufacturing industry and described its organization. Then we defined system structure and characteristics of CMEE. The idea of managing this system and transforming it into a highly autonomous, dynamic, agile, adaptive and aligned network certainly provide an appealing vision. The information and material flows are making this vision partially realizable.

However the inherent complexity of CMEE makes the efficient utilization of the information and material flows very difficult. We realized that handling this complexity has been beyond the
capabilities of the existing tools and techniques. Therefore, we emphasize in this research that in order to effectively understand a CMEE, it should be presented as a CAS. As a result we laid down some significant ideas for the extension of modeling and analysis of CMEE using the concepts, tools and techniques arising in the study of CASs and network theory. We modeled the CMEE using an agent-based simulation approach. We reformulated the computer manufacture enterprises as agents, computer manufacture enterprise market as the environment that consists of these agents and both together form the complex adaptive CMEE.

We verified the feasibility and validated usefulness of the proposed techniques. We determined the main and interactive effect of factors on the performance measures. After that, we compared possible system improvement alternatives for the multi-agent model.

This study helps us to fully understand system basics of CMEE, so that policy makers and business holders can re-model their current system structure to construct a triple-A alternative. A triple-A alternative minimizes the risk of failure and maximizes adaptivity to rapid changes, alignment with market collaborators, agility and robustness of the enterprise system.

In such an alternative model, the enterprises are empowered with required skills to evolve and self-organize. Moreover they build intelligence through interactions with other enterprises. As a result, they will be able to anticipate changes in the system quickly, synchronize with the new norms, operate profitably in a competitive environment, and survive environmental disturbances.
Finally as a result of this dissertation study, in the next section, we generated conclusions and recommendations to policy makers and business holders for handling existing complexities in the CMEE.

11.1. Summary of Findings

Twelve major contributions of this research work are as follows:

- Compiled a thorough review—which covers 189 technical articles, books, and case studies—on supply chain management, agile production systems, complex adaptive systems, network theory, organizational behavior, consumer behavior and computer manufacturing enterprise business market.

The voids in the existing literature on computer manufacturing enterprise business market and its customer-enterprise relations, enterprise-enterprise interactions and supply chain are identified.

A careful look at the voids shows that this is an emerging area with many open avenues for research. Particularly there is a large scope for research into modeling of agile, adaptive, aligned enterprise systems.

- Explored the meaning and significance of a triple-A supply chain characteristics: agility, adaptivity and alignment. Through the interpretation of Hau Lee (2004), the
triple-A supply chain concept is investigated. It is concluded that the triple-A is indeed a new supply chain paradigm which is converted to a triple-A enterprise paradigm in this dissertation.

- Interpreted a variety of strategic terms such as agency, self-organization, emergence, network connectivity, adaptivity, cooperation, non-linearity, and non-random future in the context of computer manufacturing enterprise ecosystems.

- Utilized representations (supplier and customer agents), metrics (adaptivity, agility, and alignment), algorithms (shortest path length, clustering coefficient, giant component), and methods (complex adaptive systems, network theory, agent based simulation) to realize robust CMEE that can convert a change in the business environment into a value proposition.

Moreover, the proposed work showed that enterprise systems exhibit self-organization behavior – in the sense, they operate adaptively, evolve continuously, and transform into new units over a long period of time.

- Developed a structured framework for designing a triple-A computer manufacturing enterprise ecosystem. In order to simulate various facets of CMEE, we constructed an agent based simulation that contributes to the understanding and building of agile, adaptive and aligned computer manufacturing enterprises. This simulation model also helps us to find enterprise architectures which bestow the right balance between being
rigid and being conducive to modularization and self-organization by using the agent based approach.

- Presented an in depth analysis of adaptivity and cooperation of computer manufacturing enterprises in CMEE. The results showed that adaptive enterprises, which adjust goals and infrastructure quickly according to the changes in the customers, suppliers, and/or competitors, survive longer in the CMEE compared to resistant enterprises.

According to results, enterprises, which adopt the cooperative behavior, benefit from increased collaboration and information flow with their neighbors and gain advantage over their greedy competitors. Their fitness levels increase and as a result they survive longer in the CMEE.

- Showed that companies, which are on the top of the vertical integration, are stronger in the market and survive longer than the enterprises which are away from the main customer. We also investigated the supplier behavior with respect to changing reproduction threshold for three different environmental conditions. It was shown that the more difficult to get into market and branch out (high reproduction threshold value) the fewer the numbers of enterprises survive in the CMEE.

- Investigated the effects of three different environmental conditions: the good economic times, normal economic times, and severe economic times on enterprise
ecosystem survival. The statistical results showed that there is a significant deviation in market life expectancy according to current environmental conditions of the market. It was observed that the customers’ purchasing power decreases when environmental conditions get harsher, since customers with a limited budget are willing to spend less money.

As a result the customer demand for products decreases, which consequently decreases the sales. These also cause a decrease in the product price to vitalize product sales. Decreasing sales forces companies to decrease the production rate so that they order fewer raw materials and keep fewer inventories.

Since both sales and production rate decrease, energy and budget of companies decrease significantly. Those results showed that one negative effect in the market creates a domino effect and affects every layer of the supply chain.

- Presented design of experiments results which indicate that there is a significant decrease in market life expectancy of the CMEE when the current environmental condition changes from good to harsh, reproduction threshold (decreasing enterprise ability of growing and branching out in the CMEE) increases from low to high and combination of these two factors at a harsh environmental condition and high reproduction threshold levels.
Built a network representative of the triple-A CMEE and applied the network theory to analyze features and structure of the CMEE. We observed short path length in the computer manufacturing enterprise network. It is observed that with increasing clustering coefficient the market starts to grow and with decreasing clustering coefficient the market begin to shrink.

In the light of these observations, we concluded that when computer manufacturing enterprise market is on a growing trend, it has a short path length and high tendency for clustering (reaching clustering coefficient values up to 0.967) in the computer manufacturing enterprise network which indicates the fact that while growing, CMEE turns into a small world.

The current project also strongly supports the global compatibility and competitiveness of US industries. We showed that the use of complex adaptive systems and network theory allows timely diagnosis and detections of abnormalities in the operations of a CMEE.

As a result computer manufacturing enterprises, which are agile and robust to uncertainties and rapid changes, function well in a dynamic business environment. Therefore building a triple-A CMEE establish an agile and robust profile for US computer manufacturing companies that are well prepared to global changes.
The project offers important and valuable contributions and benefits to both policy makers and companies. For policy makers the outcomes of the project helps to (1) formulate the next-generation reconfigurable computer manufacturing enterprises that meet the rapidly changing market and industry requirements better; (2) gain better understanding of evolution and organization of CMEE; (3) identify the requirements of an agile and robust computer manufacturing enterprises.

For companies the project outcomes helps to (1) understand and practice all the items stated for policy makers; (2) guard against customer demand for individualized and customized products at lower costs; (3) guard against predictable changes created by factors such as technology advancements, social trends, and environmental concerns; (4) guard against unpredictable changes such as natural calamities, war threats, and technological breakthroughs; (5) offer high-quality mass customized products at low cost with utmost time and delivery flexibility; and (6) secure a competitive edge.

Therefore the proposed work helps policy makers and business stakeholders to understand the dynamics of the formation, adaptation and evolution of CMEE that set the rules of survival and growth in the business environment.
11.2. Future Work

There are many research issues that need to be addressed before complex adaptive systems and network theory become an applied science in the fields of organizational decision making and supply chain management.

- The methods and techniques used in this dissertation need to be applied to real-world problems using real-world data to validate them.

- In this dissertation, a set of fourteen complex adaptive system features are introduced for CMEE; among them, only ten features are applied to CMEE. A further research to develop metrics for other features – dimensionality and feedback, rugged landscape, difficulty to determine boundaries and nested structure – can be attempted.

Further, we borrowed self-organization, adaptation and cooperation concepts from biological ecosystems and agility and alignment concepts from manufacturing systems. As the research in enterprise systems extends, more features can be introduced and investigated to assist better designs of business markets.

- The methodology presented in this dissertation for designing a simulation framework to better understand the possible evolutionary paths of CMEE can further be extended and tested on other enterprise ecosystems.
• In this dissertation an implementation methodology for incorporating complex adaptive systems and network theory is put into practice. However, incorporation of these techniques with other evolutionary and heuristic techniques is not explored.

• The emerging technologies such as digital product definition, Internet-aided design, virtual manufacturing, one-to-one marketing, and Internet-based electronic commerce are the key technologies that show promise to support agile, adaptive and aligned enterprises systems, which are the essential characteristics of the triple-A supply chain concept. There is an enormous scope for further research in these technology areas. These issues can be addressed in the future research.

In closing, the efforts of describing and formulating the inner workings of an enterprise ecosystem, application of agility, adaptivity and alignment concepts is highly context dependent. Therefore what an enterprise must do in order to become agile, adaptive and aligned depends on its own understanding of its customers, markets, competitors, products, competencies, and resources. Triple-A measures are not an end-goal an enterprise would reach, after which it thrives on its established capabilities; rather these concepts are means for managing change and adaptation of internal practices and external relationships to new customer opportunities, and exploiting new technologies on ongoing basis. Triple-A enterprises share infrastructure cost and business risk, core competencies, market access, product loyalty, to reduce concept-to-consumption time and to look bigger, and to become a value-based solution providers. Therefore the triple-A paradigm which embraces economies
of scope and economies of scale is emerging. Soon the triple-A enterprise concept will mark the culmination and become the synthesis of effective solutions for enterprise ecosystems.
References


References


http://www.nist.gov/el/.


References


Shi, X., B. Tseng, et al. (2009). Information Diffusion in Computer Science Citation Networks. The International Conference on Weblogs and Social Media. San Jose, CA.


Appendix A: Submodels

11.3. A.1. Observer Submodels

setup Submodel: Set-up procedure initializes the agent based world. It initializes the global variables and call main procedures to make agent based world ready to run. First of all, it calls the clear-all method, which is a built-in function in NetLogo that resets the world to an initial, empty state. All the patches turn to black and any turtles you might have created disappear. Basically, it wipes the slate clean for a new model run. Then, it initializes global variables:

- It sets particle-inertia to 0.5
- It sets attraction-to-personal-best to 2
- It sets attraction-to-global-best to 1
- It sets particle-speed-limit to 10
- It sets infinity 99999
- It sets shopped? to false
- It initializes mylist list
- It initializes child-suppliers list

And it calls procedures:

- setup-nodes
- setup-spatially-clustered-network
- find-density
- find-all-components
Appendix A: Submodels

- nodes-giant-component
- pso-supplier
- update-plot

It also asks initial-adaption-size of suppliers to become adapted via a call to become-adapted procedure. It asks all suppliers to set “collaborated_before?” to true and if cooperative-on? is equal to false to change their colors to white.

Finally it checks the global variables “color-adaptation?” and “cooperative-on?” If both of them are false then calls the procedure color-giant-component.

**setup-nodes Submodel:** This submodel calls the m-create-suppliers and m-create-costumer procedures.

**m-create-costumer Submodel:** This submodel creates “num” customers randomly distributed to virtual world with their default shape is set to "person". It sets the customer attributes; money to a random number between 0 to 99, salary to a random number between 1 to 5, and customer-type to a random number between 1 to 3. It gives the customers normally distributed random initial velocities for both x and y directions with mean 0 and standard deviation 1. It sets starting point as the customer's current best location, and personal best value, “personal-best-val”, to 0, since customers haven’t started shopping yet. Then it colors the customers according to their type.

- If customer-type is 1, then color is set to pink.
• If customer-type is 2, then color is set to orange.

• If customer-type is 3, then color is set to yellow.

**m-create-suppliers Submodel:** This submodel creates “num” suppliers randomly distributed to virtual world, with type “s-type”. Initially all suppliers set to be “not adaptive” and “not resistant”, therefore in the beginning everyone is neutral to changes. Suppliers own many attributes, six of which are link-count (0), budget (100), inventory (100), supply (50), production ((random 10) + 5) and energy (metabolism * 4), are initialized at this step.

Each supplier can compete cooperative or greedy in the market. This characteristic is set with a random probability. If the random probability is smaller than the global variable cooperative-probability then supplier competes greedy and if color-adaptation is off and cooperative-on is on then its color is set to red. Otherwise, supplier is cooperative in the market and similarly if color-adaptation is off and cooperative-on is on then its color is set to blue.

Finally, remaining attributes of each supplier are set according to their type.

• If supplier is type zero, then its shape set to building store and its product-price set to a random number between 10 and 19.

• If supplier is type one, then its shape set to filled circle and its product-price set to a random number between 5 and 14.

• If supplier is type two, then its shape set to empty circle and its product-price set to a random number between 2 and 6.
setup-spatially-clustered-network Submodel: This procedure creates a specially clustered network. It starts with initializing the loop counter, num-links, to average-node-degree multiplied by number-of-nodes. Then it loops num-links times and connects suppliers randomly vertically and horizontally. So, if there are 5 suppliers and their average degree is 4, then the loop counter, num-links, is set to 10. Then the suppliers are linked as follows:

1. **Tier 0 to tier 1**: Tier 0 suppliers, which have links less than average-node-degree, ask to link with a randomly selected tier 1 supplier that is not a link neighbor with themselves and has links less than “average-node-degree – 1”. If there exists a tier 1 supplier with these properties, they are connected with a link. Both tier 1 and tier 0 suppliers’ number of links variable, “link-count”, is increased by one.

2. **Tier 1 to tier 2**: Tier 1 suppliers, which have links less than “average-node-degree – 1”, ask to link with a randomly selected tier 2 supplier that is not a link neighbor with themselves and has links less than average-node-degree. If there exists such a tier 1 supplier they are connected with a link. Both tier 1 and tier 2 suppliers’ number of links variable, “link-count”, is increased by one.

3. **Tier 0 to tier 0**: Tier 0 suppliers, which have links less than average-node-degree, ask to link with a randomly selected tier 0 supplier that is not a link neighbor with themselves. If there exists such a tier 0 supplier they are connected with a link. Both tier 0 suppliers’ number of links variable, “link-count”, is increased by one.

4. **Tier 1 to tier 1**: Tier 1 suppliers, which have links less than average-node-degree, ask to link with a randomly selected tier 1 supplier that is not a link neighbor with themselves.
If there exists such a tier 1 supplier they are connected with a link. Both tier 1 suppliers’ number of links variable, “link-count”, is increased by one.

5. **Tier 2 to tier 2**: Tier 2 suppliers, which have links less than average-node-degree, ask to link with a randomly selected tier 2 supplier that is not a link neighbor with themselves. If there exists such a tier 2 supplier they are connected with a link. Both tier 2 suppliers’ number of links variable, “link-count”, is increased by one.

Finally, to make the network look prettier, the layout-spring algorithm is used to place the suppliers and set the link distances.

**go Submodel**: This procedure first sets giant component size of the supplier network, giant-component-size, to zero. Then it calls the following procedures:

- face-effects-of-env
- go-procedures
- do-statistics (if number of links in the supplier network is greater than one)
- pso-supplier
- pso-customer
- balance
- tick
- update-plot

Also, it states the stop condition for the simulation. If number of any type of suppliers or customers is less than equal to zero, then simulation halts.
**go-procedures Submodel:** In this procedure, first of all, observer asks customer agents to call do-shopping submodel, and then increase their money in the amount of their salary. Then two global variables are initialized. The “supply-available?” variable is set to true and the list elements of “child-suppliers[]” list are initialized to 0.

Next, four local variables; current demand (temp-demand), expected product price (expected-pprice), current product price (temp-pprice) and current production level (temp-production) are created and set equal to zero. Then it calls the suppliers as follows:

*Supplier type 0 (main suppliers):* It asks to type 0 suppliers to set their attributes and local variables

- temp-demand to their demand and
- expected-pprice to their product-price.

Then it first checks if they have any link neighbors of type 1 suppliers. If not, they select one of the type 1 suppliers which has links less than “average-node-degree + 2” and they create a link with that supplier. They increase their and the linked supplier’s number of links attribute by one.

The next thing the type 0 suppliers do is to check if there are better suppliers which are type 1 in the network. A better supplier criterion is defined by wealth of enterprises. If they find a better
choice, they drop their weakest link neighbor (supplier type 1) and they link with the better choice.

After that type 0 suppliers satisfy their common tasks such as full-filling customer demand, buying new inventory, updating their budget, supply and inventory, check their fitness values for survival, check their fitness values for opening new stores (reproduce) and propose to merge with competitors.

A type 0 supplier knows its demand through do-shopping procedure. When it sells “demand” number of items from its supplies, it has to replace the sold items by purchasing new inventory from supplier type 1 and producing new items. So it asks suppliers which are link neighbors and of type 1 whether they are available, have enough supply and affordable product price (product-price $\leq$ expected-pprice) or not. If they satisfy all three criteria, they become unavailable for other suppliers, their attributes

- demand set to temp-demand (temp-demand is type 0 supplier’s demand),
- supply set to previous supply minus demand so that we decrease sold items from inventory,
- budget set to previous budget plus temp-demand times product-price so that we increase money in budget by number of sold products times their prices and
- temp-pprice set to product-price so that we hold product-price of sold items in a temporary variable.
After purchasing new inventory, if it has more than a certain amount of inventory it produces/fabrics new supplies in production rate, if not it produces new supplies less than the normal production rate.

Then the supplier type 0 updates its budget, supply and inventory according to the criterion “if it can purchase new supplies from supplier type 1” or “not”. Also it sets “supply-available?” to true.

Then it calls the following procedures:

- reproduce (reproduce returns the newly created supplier and it is added to list “child-suppliers”)
- calculate-energy
- merge

**Supplier type 1 (Tier 1 suppliers):** Observer asks to type 1 suppliers to set their attributes and local variables

- temp-demand to their demand and
- expected-price to their product-price.

Then Tier 1 suppliers first check if they have any link neighbors of Tier 2 suppliers. If not, they select one of their Tier 2 suppliers which has links less than “average-node-degree + 2” and they
create a link with that supplier. They increase their and the linked supplier’s number of links attribute by one.

Similar to the type 0 suppliers, next think the type 1 suppliers do is to check if there are better suppliers which are type 0 or type 2 in the network. If they find a better choice, they drop their weakest link neighbor (supplier type 0 or supplier type 2) and they link with the better choice. Type 1 suppliers check links both with type 0 and type 2 suppliers since they are in the middle of vertical integration.

After that type 1 suppliers satisfy their common tasks such as full-filling customer demand, buying new inventory, updating their budget, supply and inventory, check their fitness values for survival, check their fitness values for opening new stores (reproduce) and propose to merge with competitors.

A type 1 supplier knows its demand through type 0 suppliers that buys its products. When it sells “demand” number of items from its supplies, it has to replace the sold items by purchasing new inventory from supplier type 2 and producing new items. So it asks suppliers which are link neighbors and of type 2 whether they are available, have enough supply and affordable product price (product-price <= expected-price) or not. If one of the link neighbor supplier type 2 satisfies all three criteria, it becomes unavailable for other suppliers, its attributes

- demand set to temp-demand (temp-demand is type 1 supplier’s demand),
Appendix A: Submodels

- supply set to previous supply minus demand so that we decrease sold items from inventory,
- budget set to previous budget plus temp-demand times product-price so that we increase money in budget by number of sold products times their prices and
- temp-price set to product-price so that we hold product-price of sold items in a temporary variable.

After purchasing new inventory, if type 1 suppliers have more than a certain amount of inventory they produce new supplies in production rate, if not they produce new supplies less than the normal production rate.

Then the supplier type 1 updates its budget, supply and inventory according whether if it can purchase new supplies from supplier type 2 or not. Also it sets “supply-available?” to true.

Then it calls the following procedures:

- reproduce (reproduce returns the newly created supplier and it is added to list “child-suppliers”)
- calculate-energy
- merge

Supplier type 2 (Tier 2 suppliers): Observer asks to type 2 suppliers to check if they have any link neighbors of type 1 suppliers. If not, they select one of their type 1 suppliers which has links
less than “average-node-degree + 2” and they create a link with that supplier. They increase their and the linked supplier’s number of links attribute by one.

Next think the type 2 suppliers do is to check if there are better suppliers which are type 1 in the network. If they find a better choice, they drop their weakest link neighbor (supplier type 1) and they link with the better choice.

After that type 2 suppliers satisfy their common tasks such as full-filling customer demand, buying new inventory, updating their budget, supply and inventory, check their fitness values for survival, check their fitness values for opening new stores (reproduce) and propose to merge with competitors.

The suppliers/ resource of type 2 suppliers are not modeled in this simulation; therefore, in each iteration, their inventory increases with a constant number. After purchasing new inventory, if they have more than a certain amount of inventory they produce new supplies in production rate and decrease the number of inventories accordingly, if not they produce new supplies less than the normal production rate and update the inventory.

Then the supplier type 2 updates its budget and supply. Then it calls the following procedures:

- reproduce (reproduce returns the newly created supplier and it is added to list “child-suppliers”)
- calculate-energy
After all customers and supplies are completed their tasks for the current iteration, observer loops over “child-suppliers” list elements and calls the make-newcomer function to create the new companies who should join to the system as a result of reproduction submodel. In addition to that it creates new companies with random probability p.

The observer asks all suppliers, if their link count is greater than one. The isolated suppliers die before next iteration. Then the observer asks all suppliers to compete via calling compete submodel.

After that, observer checks a terminating condition. If all the suppliers are adaptive to changes then it calls lose-adaptivity submodel which forces all suppliers to become neutral. If all the suppliers are resistant to changes then it stops the simulation.

Then if “adaptivity-option?” switch is on observer calls “start-change” and “do-change-checks” submodels else it calls “start-change-option2” submodel.

Then if number of suppliers is greater than half of number of customers then new customers are created. If number of suppliers is less than one fifth of customer, some of the customers are chosen randomly and killed.

**do-statistics Submodel:** The submodel first initializes elements of list, “mylist”, to zero. Then it calls the following submodels:
Appendix A: Submodels

- find-density
- find-all-components
- color-giant-component (Procedure is called if both “color-adaptation?” and “cooperative-on?” switches are off.)
- nodes-giant-component
- do-calculations

**update-plot Submodel:** This procedure draws the "Network Status" graph. In this graph red line presents percentage of suppliers who are neutral to changes in the market. The green line presents percentage of suppliers who are adapted. The red line presents percentage of suppliers who are resistant to changes in the market.

**Balance:** Balance function basically checks the number of each supplier type in the virtual world. If number of one type decreases to one, it asks one of the maximum numbered type supplier with minimum budget to change its type to dying out type.

**face-effects-of-env:** If environmental affects slider is set to 3, every iteration one of cooperative suppliers is chosen randomly and changed to a greedy one. Therefore number of greedy suppliers increases by time. If environmental affects slider is set to 1, every iteration one of greedy suppliers is choose randomly and changed to a cooperative one. Therefore number of cooperative suppliers increases by time.
face-effects-of-env-option2: This submodel is similar to face-effects-of-env submodel however instead of using user interface slider, it uses reporter “ticks”. If ticks reporter is less than a constant “time”, one of cooperative suppliers is chosen randomly and changed to a greedy one. Therefore number of greedy suppliers increases by time. If ticks is between “time + 1” and “time + delta”, one of greedy suppliers is chosen randomly and changed to a cooperative one. Therefore number of cooperative suppliers increases by time.

psd-customer: This submodel updates the "personal best" location for each customer, if customers have found a new value better than their previous "personal best". It also updates the "global best" location for all the customers. Customers change their velocities by being attracted to their "personal best" value and then change their velocities by being attracted to the "global best" value anyone has found so far. They face in the direction of their resulting velocity and move forward by the magnitude of it.

psd-supplier: This submodel calculates the “fitness-reliability”, “fitness-popularity”, “fitness-bestprice” and “fitness-value” of suppliers.

- fitness-reliability = supply + budget + energy
- fitness-popularity = demand
- fitness-bestprice = inverse of product-price
- fitness-value = fitness-reliability + fitness-popularity + fitness-bestprice
Appendix A: Submodels

It calls the normalize submodel and normalize all four values. Then it picks the Tier 0 supplier with maximum “fitness-value”, and assigns its “fitness-value” to global variable “true-best-supplier”.

Network Statistics Methods:

*find-all-components*: This procedure finds all the connected components in the network, their sizes and starting nodes. Also assigns the start node of giant component to a global variable.

*explore [new-color]*: This method finds all supplier agents reachable from given node and if “color-adaptation?” and “cooperative-on?” switches are off re-colors them.

*color-giant-component*: This method colors the giant component blue, if “color-adaptation?” and “cooperative-on?” switches are off.

*add_mylist [node]*: This method finds all agents reachable from given node and adds them to a list (one dimensional array).

*nodes-giant-component*: This method explores all supplier agents that are in the giant component and adds those agents to a list.

*do-calculations*: This procedure reports true if the network is connected, and reports false if the network is disconnected. (In the disconnected case, the average path length of network does not
make sense, or perhaps may be considered infinite, therefore in that case it uses the giant component for the calculations.) It calculates average-path-length and then finds the clustering coefficient and adds to the aggregate for all iterations and also finds the clustering coefficient of giant component and adds to the aggregate for all iterations.

find-density: This procedure calculates the density of the supplier network.

in-neighborhood? [hood]: This procedure decides whether the supplier agent is in the neighborhood of two other supplier agents. This method is used in clustering coefficient calculations.

find-clustering-coefficient: This procedure finds the clustering coefficient of all supplier networks.

find-clustering-coefficient-giant: This procedure finds the clustering coefficient of giant component in supplier network.

find-path-lengths: This procedure finds path length from each supplier to every other supplier.

11.4. A.2. General Agent Submodels (general to any agent type)

normalize: This submodel takes three inputs; value, maximum value, minimum value. It normalizes the variable value according to maximum and minimum values.
11.5. A.3. Supplier Submodels

**make-newcomer:** It creates one new supplier on an empty place (patch) on earth which is adaptive, has 100 items in its inventory, has a production rate of 5 to 10 items per tick and doesn’t have any collaboration with current suppliers before. The new supplier is attached to an already existing supplier with \% 50 chance of making random attachment to the closest neighbor or \% 50 chance of making preferential attachment.

**find-partner:** This code is borrowed from Lottery Example (from the Code Examples section of the NetLogo Models Library). The idea behind the code is a bit tricky to understand. Basically it takes the sum of the degrees (number of connections) of the suppliers. That's the number of "tickets" in the lottery. Then a random "ticket" (a random number) is picked. Then the algorithm steps through the suppliers to figure out which supplier holds the winning ticket.

**become-adapted:** This procedure sets adapted attribute to true, resistant attribute to false and color attribute to green of a supplier.

**realize-change-in-market:** This procedure sets adapted attribute to false, resistant attribute to false and color attribute to red of a supplier.

**become-resistant:** This procedure sets adapted attribute to false, resistant attribute to true and color attribute to grey of a supplier.
**start-change:** If a supplier is adaptive, this method changes its neighbors, which are neutral, to adaptive (red to green); if random number generated is smaller than chance-of-change.

**start-change-option2:** This submodel uses totalistic rules for adaptation instead of the stochastic rule that is used in submodel start-change.

**lose-adaptivity:** If a supplier is adaptive, this model changes it to neutral state (green to red) with a random probability.

**do-change-checks:** If the random number picked smaller than lose-interest-to-change, the algorithm picks a new random number. If the new random number is smaller than resist-change, it changes the corresponding supplier to resistant (green to grey) otherwise it changes it neutral (green to red).

**%inventory:** This submodel finds the average inventory of all suppliers in the system.

**avg-product-price:** This submodel calculates average product price of all suppliers.

**avg-energy-suppliers:** This submodel calculates average energy of all suppliers.

**avg-budget-suppliers:** This submodel calculates average budget of all suppliers.
**avg-demand-suppliers:** This submodel calculates average demand of all suppliers.

**calculate-energy:** This submodel calls basal-metabolism submodel and then calculates energy of each supplier.

**basal-metabolism:** This submodel calculates the basal metabolism of each supplier in every tick according to environmental effects. If environmental effects variable, “env-effects”, is equal to one, then it decreases supplier’s energy by two. If “env-effects” variable is equal to two, then it decreases supplier’s energy by four. Otherwise, it decreases supplier’s energy by eight.

Then it checks several termination conditions. If energy or budget of supplier less than zero or supply or inventory is less than a constant number, it removes the supplier from the simulation. After the removal, it decreases the neighbors’ link counts by one. Also, if the removed supplier is the start node of giant component, it chooses one of link neighbors and assigns it as the start node of giant component.

**compete:** If supplier competes cooperative, this method calls compete-cooperative submodel, otherwise it calls compete-greedy submodel.

**compete-cooperative:** If product price of supplier is less than average product price, the supplier increases its product price by one so that it aligns its product price with other companies.
**compete-greedy**: If product price of supplier is greater than average product price, the supplier decreases its product price by two so that it has a lower product price than its competitors.

**reproduce**: If the supplier’s energy is greater than the reproduction threshold, its energy decreases by reproduction cost. Then the supplier itself is returned from this function to be used in make-newcomer submodel.

**merge**: This submodel models the merging of two companies. First it creates and initializes several local variables as follows:

- let my-budget budget
- let my-energy energy
- let my-supplier-type supplier-type
- let choice self
- let n-link-count 0
- let n-budget 0
- let n-inventory 0
- let n-supply 0
- let n-production 0
- let n-energy 0
- let n-cooperative? false
- let n-adaptive? false
Then if the supplier’s budget, which is firing this method, is greater than average budget of suppliers and if its energy is greater than average energy of suppliers, it chooses one of its link neighbors, which has the same type with itself, and assigns it to the variable “my-choice” meaning that it is a horizontally integrated supplier. If there is such a neighbor, it compares its budget and its energy with “my-choice”. If both the attributes are greater than “my-choice” attribute values, it writes “my-choice” attribute values to local variables

- set n-link-count link-count
- set n-budget budget
- set n-inventory inventory
- set n-supply supply
- set n-production production
- set n-energy energy

If “my-choice” is cooperative it becomes cooperative. If “my-choice” is adaptive, it becomes adaptive. Then it links with all neighbors of “my-choice”. If “my-choice” is the start node of giant component, it becomes the start node of giant component. After all it removes “my-choice” from the virtual world. Then it updates its own attributes as follows:

- set link-count n-link-count + link-count
- set budget n-budget + budget
- set inventory n-inventory + inventory
- set supply n-supply + supply
- set production n-production + production
• set energy (n-energy / 2) + energy


do-shopping: If customer is on the same patch with a supplier and if supplier has enough inventories, costumer buys one item, and custemr’s money decreases by one.

%money: Tracks average money each costumer has in the market.

income [ customer shop ]: The submodel reports the number of products that are purchased by a customer.

• If the customer is a limited income customer and the supplier sells products less than or equal to average market price, the customer buys two items.
• If the customer is a limited income customer and the supplier sells products greater than average market price, the customer buys one item.
• If the customer is not a limited income customer and the supplier sells products less than or equal to average market price, the customer buys four items.
• If the customer is a limited income customer and the supplier sells products greater than average market price, the customer buys three items.
Appendix B: Interface

11.7. B.1. Buttons

There are two types of buttons in the user interface of the simulation model. First button, *Setup*, is an ones-only button which calls setup submodel which resets the world to an initial, empty state. Then it initializes the agent based world; patches, turtles and links, initializes the global variables and call main procedures to make agent based world ready to run. Second button, *Go*, is a forever button that calls go submodel. It executes the instructions given in the go submodel over and over, until user clicks on the button again to stop the action or a termination condition is met.

11.8. B.2. Sliders

*initial-adaptation-size* is a slider which determines the number of suppliers that are initially adaptive.

*number-of-nodes* determines the initial number of suppliers and customers in simulation world.

*average-node-degree* is the average number of links a supplier has.

*chance-of-change* is the probability that determines being adaptive in the market.

*lose-interest-to-change* is the probability that determines being neutral to changes in the market.

*resist-change* is the probability that determines being resistant to changes in the market.
Appendix B: Interface

\( p \) a number between 0 and 1, which is used for stochasticity in the model.

*metabolism* is the rate of basal metabolism of a supplier.

*cooperative-probability* is the probability that determines being cooperative in the market.

*reproduction-cost* is the cost of reproduction, opening a new branch, factory, etc. in the market for each individual supplier.

*reproduction-threshold* is the threshold value above which a supplier reproduces.

*limited-income* is the value that determines the purchase power of a customer.

*env-effects* is the slider which determines the market conditions. 1 represents nice conditions, 2 represents normal conditions, and 3 represents severe conditions.

11.9. B.3. Switches

*adaptivity-option*? If this slider is on, this method calls the adaptation property methods. Otherwise, the adaptation is ignored in the model.

*color-adaptation*? If this slider on, we can observe the adaptation on the network. Adaptive suppliers are colored to green, neutral suppliers are colored to red and resistant suppliers are colored to grey.
cooperative-on? If this slider is on and “color-adaptation?” slider is off, we can observe the cooperative and greedy behavior in the network. Cooperative suppliers are colored to blue and greedy ones are colored to red. If both “color-adaptation?” and “cooperative-on?” sliders are off, then we can observe the giant component of network. Nodes that are in the giant component are colored to blue.

spring-layout  If this slider is on, the “layout-spring” built-in NetLogo function is called which arranges the suppliers in turtle-set, as if the links in link-set are springs and the suppliers in turtle-set are repelling each other.

11.10.  B.4. Input

type-0: The number of initial main suppliers.

type-1: The number of initial Tier 1 suppliers.

11.11.  B.5. Monitor

# of customers: Shows number of customers in each simulation tick.

# of suppliers: Shows number of suppliers in each simulation tick.

#type0: Shows number of main suppliers in each simulation tick.

#type1: Shows number of Tier 1 suppliers in each simulation tick.
#type2: Shows number of Tier 2 suppliers in each simulation tick.

average-money: Shows average money that customers have in the market in each simulation tick.

average-inventory: Shows average inventory that suppliers have in the market in each simulation tick.

avg-product-price: Shows average product price that suppliers have in the market in each simulation tick.

avg-demand: Shows average demand that suppliers have in the market in each simulation tick.

avg-energy: Shows average energy that suppliers have in the market in each simulation tick.

avg-budget: Shows average budget that suppliers have in the market in each simulation tick.

density: Shows density of network in each simulation tick.

giant-component-size: Shows giant component size of suppliers in network in each simulation tick.

average-path-length: Shows average path length of suppliers in network in each simulation tick.

clustering-coefficient: Shows clustering coefficient of suppliers in network in each simulation tick.

c-coeff-giant: Shows clustering coefficient of suppliers in giant component in each simulation tick.