論文 / 著書情報
Article / Book Information

題目(和文)	エージェント・シミュレーションとサービス・ドミナント・ロジックに基づくカスタマー・ロイヤルティの動的変動に関する研究
Title(English)	Study on the Dynamics of Customer Loyalty from the Service-Dominant Logic Perspective using Agent-Based Simulation
著者(和文)	RajapakseChathura
Author(English)	Chathura Rajapakse
出典(和文)	学位:博士(工学), 学位授与機関:東京工業大学, 報告番号:甲第9870号, 授与年月日:2015年3月26日, 学位の種別:課程博士, 審査員:寺野　隆雄,新田　克己,高安　美佐子,出口　弘,小野　功
Citation(English)	Degree:Doctor (Engineering), Conferring organization: Tokyo Institute of Technology, Report number:甲第9870号, Conferred date:2015/3/26, Degree Type:Course doctor, Examiner:,,,,
学位種別(和文)	博士論文
Type(English)	Doctoral Thesis

Study on the Dynamics of Customer Loyalty from the Service-Dominant Logic Perspective using Agent-Based Simulation

(エージェント・シミュレーションとサービス・ドミナント・ロジックに基づくカスタマー・ロイヤルティの動的変動に関する研究)

Doctoral Thesis By

Chathura Prasanna Rajapakse

チャトゥラ　プラサンナ　ラージャパクセ

Supervised by

Professor Takao Terano
寺野 隆雄 (教授)

Tokyo Institute of Technology

Interdisciplinary Graduate School of Science and Engineering
Department of Computational Intelligence and Systems Science

Tokyo, Japan

March 2015

Dedicated to the name of my father
(1943 ~ 2010)

ACKNOWLEDGEMENT

First and foremost, I would like to express my heartiest gratitude to my academic advisor, Professor Takao Terano, not only for his guidance for this thesis but also for the support and guidance extended throughout my entire academic life at the Department of Computational Intelligence and Systems Science, Tokyo Institute of Technology, Japan. He immensely contributed to make me a competent researcher ready to take future challenges in making the world a better place.

Secondly, I thank my panel of examiners for their valuable comments and very constructive criticisms, which helped me a lot to improve this thesis. Furthermore, all academic and non-academic members of the Department of Computational Intelligence and Systems Science must be kindly remembered for their dedication and commitment to maintain a high quality atmosphere for academic research. Furthermore, I extend a big thank to Dr. Yamada, Ms. Yamaguchi and all members of the Terano laboratory for making the laboratory a pleasant place for me to stay for a long period in my life.

I must also thank the Ministry of Education, Culture, Sports, Science and Technology (MEXT) in Japan for offering me a scholarship to study at Tokyo Institute of Technology for my doctoral degree. MEXT making arrangements to offer scholarships for foreign students, especially those from developing countries, is absolutely commendable from the perspective of socio-economic development of less developed regions of the world.

Last but not least, I thank my family, especially my wife Anusha, daughter Desandi and son Thehas, for being so patient during the past couple of years.

DECLARATION

I hereby declare that I am the sole author of this thesis, which is a copy of the final version accepted by the panel of examiners.

Chathura Prasanna Rajapakse

TABLE OF CONTENTS

ABSTRACT iii
LIST OF FIGURES v
LIST OF TABLES vii

1. Introduction 1

1.1. Background 1
1.2. Motivation and Research Questions 2
1.3. The Service-Dominant Logic Perspective 3
1.4. Methodology 4
1.5. Objectives 4
1.6. Thesis Contents 5

2. Literature Review 6

2.1. Markets as Systems 6
2.1.1. Service-Dominant Logic 7
2.1.2. Service Science 11
2.2. Customer Satisfaction, Engagement and Loyalty 13
2.2.1. Measures of Customer Satisfaction 14
2.2.2. Role of Customer Engagement in Dynamics of Loyalty 15
2.3. Agent-based Modeling and Artificial Markets 18
2.3.1. The Discipline of Agent-based Modeling 18
2.3.2. Artificial Market Research 19
2.3.3. Kauffman's NKCS Architecture 20

3. The Proposed Agent-based Model 23

3.1. Overview of the Model 24
3.1.1. Purpose 24
3.1.2. Entities, State Variables and Scales 24
3.1.3. Process Overview and Scheduling 33
3.2. Design Concepts 36
3.2.1. Basic Principles 36
3.2.2. Emergence 45
3.2.3. Adaptation 45
3.2.4. Objective of Adaptation 45
3.2.5. Learning 46

3.2.6. Sensing of Agents 46
3.2.7. Predictions 46
3.2.8. Interactions 47
3.2.9. Stochasticity 47
3.2.10. Observation Data 47

3.3. Details 47

3.3.1. Initialization 47
3.3.2. Input Data 48
3.3.3. Sub Models 48

4. Simulation Experiments and Results 59

4.1. Dynamic Behavior of Customer Loyalty 59

4.1.1. Dynamics of Loyalty and Customers' Switching Behavior 60
4.1.2. Robustness of the Result 62

4.2. Sensitivity to Competition 63

4.2.1. Early Vs. Late Competition 63
4.2.2. Impact of Expectation Growth 66
4.2.3. Impact of Loyalty Biasness 68

5. Discussion 70

5.1. About the Model 70

5.1.1. What this Model Is? 70
5.1.2. Uses of the Model 71
5.1.3. What this Model Is Not? 73

5.2. Potential Application – Tourism Sector of Sri Lanka 74

5.2.1. Why Tourism Domain? 74
5.2.2. Sri Lanka as a Tourism Market 75
5.2.3. Application Guidelines to a Tourism Market 77

6. Conclusion 83

6.1. Remarks on Research Questions 83
6.2. Implications of the Research 84
6.3. Limitations 85
6.4. Future Work 85

6.4.1. Long-term Future Work 85
6.4.2. Short-term Future Work 86

References 87

Appendix A - List of Key Publications 93

Appendix B - Conference Presentations 94

Appendix C – Lists of Methods in Classes 95

Appendix D – Extended Pseudo Codes of Major Processes 99

ABSTRACT

Advancement of technology has enabled new forms of market interactions as well as faster information exchange among market actors. For example, the electronic social media act as means of customer-provider and customer-customer relationships, facilitating constructive discussions and exchange of recommendations. In other words, modern markets could be represented as systems comprising multiplicities of customers, service providers and other kinds of stakeholders who rapidly interact with each other in the forms of services, information exchange, recommendations, etc.

Customer loyalty has traditionally been recognized as playing a major role in customer revisits, except in the case of fake loyalty caused by lock-in situations. Therefore, in the quest for an answer to the traditional question of "why customers switch", customer loyalty has been studied extensively from different dimensions to identify possible causes of loyalty. Especially the relationship between customer satisfaction and loyalty has been thoroughly studied and quite prevalent (despite criticisms). However, the modern research on customer loyalty is more directed towards customers' emotional responses towards consumption situations such as "affect" and "customer engagement". Especially since about a decade, customer engagement is getting increasing attention in the research community, both as a psychological process that drives loyalty as well as a psychological state at which loyalty is a consequence.

Notably, these research studies pay less attention to the possible dynamics of loyalty stemming from interactions among market actors in modern markets. For example, an emotional response of a customer at a particular consumption situation or with a new provider just came to the market can reach hundreds of other potential customers within a short period of time. On the other hand, competitive moves of providers to delight their customers elevate the customer expectation levels, making customer satisfaction harder to all providers. Therefore, customer loyalty may need to be studied as a dynamic property in a complex system for better understanding. In other words, focusing on the time dimension of loyalty in competitive business environments would be a necessity to see how loyalty varies over time and its possible relationships with customers' switching.

A study of such system dynamics demands a systems thinking approach to capture the non-linearities and uncertainties involved with the interactions. However, the conventional research methods in business and marketing lack such a systems thinking approach as they rather focus on the properties of individual components than the interactions between

the components. This research introduces a novel method to study the dynamics of loyalty taking the systems thinking approach by combining Agent-based Modeling (ABM) and Service-Dominant (S-D) Logic.

When it comes to the "market as a system" viewpoint, the service-dominant logic unarguably has done a tremendous contribution. It views market as a system of actors, who possess resources and provide services to other actors by exchanging resources. This transcends the traditional transactional view of market interactions to a relational level. In the traditional transactional view, value is added to a product or service (the intangible product) at different points of the value chain, and exchanged for something (usually money) to complete a transaction. Thus, the traditional view considers value-in-exchange. In contrast, service-dominant logic views any tangible or intangible offering (of resources) by a service provider as a means of delivering a service, and value as being co-created at the time of use by the beneficiary by combining provider's resources with its own resources. This relational view focuses on the value-in-use than value-in-exchange. As the relationship between loyalty and value-in-use is recognized, service-dominant logic opens up a new perspective for research on customer loyalty.

Agent-based Modeling, on the other hand has got significant attention in the study of complex adaptive systems. However, no agent-based market model has been reported to date that is constructed upon the foundations of service-dominant logic. This thesis presents an agent-based market model constructed upon the foundations of service-dominant logic to study the dynamics of customer loyalty. It adopts the "service system abstraction" to define agents and Kauffman's NKCS architecture to the computational representation of agents and value co-creation in their interactions. It further defines customer loyalty as stemming from one's "affective commitment" towards a provider, which is a combination of "trust" and the strength of "peer recommendations" received about the provider. Customer agents use their loyalty values towards each service provider probabilistically to choose a provider when they get a service need. The resulting service experience is evaluated in terms of co-created value against the expectation of the customer at that time, which grows by a certain percentage at each positive service experience.

The thesis reports an extensive study of related literature, the details of the formalization of the model with respect to the literature, key outcomes as well as a discussion on the potential of applying this model to the tourism sector using Sri Lanka as the test bed.

LIST OF FIGURES

2.1	Overview of Related Work	6
2.2	The ISPAR Model of Service System Interactions	12
2.3	The Customer Engagement Process Model by Bowden	17
3.1	Organization of Model Details According to the ODD Protocol	23
3.2	Model Overview	24
3.3	Entities and their Relationships as a Class Diagram	24
3.4	Process of Selecting a Provider at a Service Need	34
3.5	The Process of Service Interaction between a Customer and a Provider	35
3.6	Illustration of the Representation of Value Proposition and Current State	37
3.7	The Dependency Structure of Value Items according to the NKCS Model	39
3.8	Algorithm that Determines Satisfaction on Attributes based on Matzler et al.'s Categorization	42
3.9	The Correspondence between Trust and Satisfaction	44
3.10	Sub Models of Building the Model Context	49
3.11	Sub Models of the Market Process	55
4.1	Dynamic Behavior of Customer Loyalty with Time	59
4.2	Number of Customers Arrived for Each Provider over Time	60
4.3	Frequency of Customer Switching over Time	61
4.4	Frequency of Customer Switching Over Time at Higher Initial Expectations	62
4.5-(a)	Change of Average Affection with a Competitor Added at Time = 50	63
4.5-(b)	Change of Market Share with Early Competition	64
4.6	Change of Average Affection with a Competitor Imitating the Best Existing Provider Added at Time = 50	65
4.7-(a)	Change of Average Affection with a Competitor Added at Time = 200	65
4.7-(b)	Change of Market Share with Late Competition	66
4.8-(a)	Change of Average Affection with Low Expectation Growth Rate (=1%) and a Competitor Added Early (time = 50)	67
4.8-(b)	Change of Market Share with Low Expectation Growth Rate (=1%) and a Competitor Added Early (time = 50)	67
4.9-(a)	Change of Average Affection with Higher Loyalty Biasness (=90%) and a Competitor added Early (time = 50)	68
4.9-(b)	Change of Market Share with Higher Loyalty Biasness (=90%) and a Competitor added Early (time = 50)	69
5.1	Comparison of Tourist Arrivals to Sri Lanka in 2013 and 2014	76
5.2	Tourism Area Life Cycle of Butler, 1980	77

5.3	Part of a Hotel's Value Proposition and Potential States	78
5.4	Part of a Tourist's Value Proposition and Potential States	79
5.5	How Value Co-creation Works in Tourism	80
5.6	Service System Interactions in a Tourism Market	81

LIST OF TABLES

2.1	Fundamental Premises of Service-Dominant Logic	8
2.2	Comparison of Goods-Dominant Logic and Service-Dominant Logic on Value Creation	9
2.3	The Five Fundamental Propositions of Customer Engagement	15
3.1	Attributes of the Entity “Value Item”	25
3.2	Attributes of the Entity “Customer”	26
3.3	Attributes of the Entity “Provider”	27
3.4	Attributes of the Entity “Service Experience”	28
3.5	Attributes of the Entity “Recommendation”	28
3.6	Attributes of the Entity “Affection”	29
3.7	Attributes of the Entity “Transaction”	29
3.8	Attributes of the Entity “Customer Relationship”	29
3.9	Attributes of the Entity “Context Builder”	30
3.10	Attributes of the Entity “Controller”	31
3.11	Model Parameters	32
3.12	Initial Parameter Values Used in the Simulation	48

1. Introduction

1.1. Background

"Why do customers switch?" or "Why do customers defect?" is an important question in business and marketing. In competitive business environments, business entities are proactively working on continuously improving their service and competence levels expecting to secure the market share and competitiveness. In this endeavor, it is advocated to put greater emphasis on retaining customers than attempting to acquire new customers, as the retained customers bring in long-term benefits to the company [1]. For example, the cost of acquiring a new customer is a one-time expense and it is not incurred with repeat customers. Instead, the repeat customers are likely to contribute to the growth of the company by creating positive word-of-mouth, extra purchases, advocacy, etc. Therefore, retaining their existing customers is an important objective of any company, and hence the question "why customers defect?" is valid all time.

While it is not always accurate to consider customer retention as a proxy to customer loyalty [1], loyalty of customers has been thoroughly acknowledged as a key determinant of repeat customer visits. According to Oliver, loyalty refers to the extent, to which customers feel committed to suppliers and do not actively seek out replacement suppliers [2]. Over the past, the determinants of loyalty have been studied taking different approaches, from the measures of mere satisfaction to the measures of emotional consumption responses while attention being paid recently to the concept of customer engagement [3]. According to Bowden, there exists a need for the development of measurement models that more effectively account for the depth of customers' emotional responses, to consumption situations. These emotional responses of customers to consumption situations are closely related to the concept of “value-in-use” found in Service Dominant Logic [4]. Therefore, Service-Dominant Logic (SD Logic) has become an important aspect in the discussion on the determinants of customer loyalty.

1.2. Motivation and Research Questions

The motivation for this research comes from the observations on the changes taking place in business environments over that past decade. Compared to the previous century, markets today have become complex systems with increased connectivity among its actors. For example, the growing popularity of electronic social networks has opened up new means for maintaining firm-customer relationships as well as customer-customer relationships. This has enabled faster exchange of information, opinions and recommendations among system actors. For example, an emotional response of one customer related to a particular consumption experience or with a new provider came to the market, can get disseminated through electronic social networks among hundreds of potential and current customers within a short period of time. Frequent exchange of such information is more likely to influence the next purchase decisions of some potential customers. On the other hand, business competitiveness has grown rapidly in the recent years with the development of technology. In most industries, firms are continuously focusing on giving delightful experiences to their customers. However, such attempts to delight customers elevate the levels of customer expectations, making customer satisfaction a challenge [6].

Therefore, this thesis holds the view that the study on customer loyalty has to take these changes in the market conditions into consideration. In other words, customer loyalty needs to be studied as a dynamic property influenced by the customers’ emotional responses in consumption situations and the rapid exchange of consumption experiences among customers. This emphasizes the necessity of adopting the systems thinking approach and focusing on the interactions between system components to understand potential dynamics of loyalty. For example, value co-creation in firm-customer interactions generates emotional responses at use contexts, varying customers' trust and involvement with respective providers and influencing their next purchase decisions and behaviors in customer forums. The significance of this approach is further corroborated with the work of Bowden where customer loyalty is conceptualized as stemming from interactive customer experiences in the process of customer engagement [3] as well as the work of Brodie et al. where customer loyalty is seen as a consequence of dynamic, iterative customer engagement processes [6].

However, apart from such conceptual models, research to date has no evidence of any research conducted to study loyalty as a dynamic property stemming from interactions in markets. Hence, this thesis is inspired by the following research questions.

- How does customer loyalty dynamically change in competitive business environments?
- How such dynamics could be effectively studied?
- If such dynamics exist, how are customers' switching decisions associated with them?

1.3. The Service-Dominant Logic Perspective

When it comes to the "market as a system" viewpoint, the service-dominant logic unarguably has done a tremendous contribution. It views market as a system of actors, who possess resources and provide services to other actors by exchanging resources [7]. This transcends the traditional transactional view of market interactions to a relational level. In the traditional transactional view, value is added to a product or service (the intangible product) at different points of the value chain, and exchanged for something (usually money) to complete a transaction. Thus, the traditional view considers value-in-exchange. In contrast, service-dominant logic views any tangible or intangible offering (of resources) by a service provider as a means of delivering a service, and value as being co-created at the time of use by the beneficiary by combining provider's resources with its own resources. This relational view focuses on the value-in-use than value-in-exchange. As the relationship between loyalty and value-in-use is recognized [6], service-dominant logic opens up a new perspective for research on customer loyalty.

1.4. Methodology

The traditional approach of studying complex systems involves decomposing the system into components to analyze the properties of components. However, such approach is not effective in the study of human systems that involve non-linearities [8]. In contrast, systems thinking approach focuses on how the thing being studied interacts with the other constituents of the system—a set of elements that interact to produce behavior—of which it is a part. This means that instead of isolating smaller and smaller parts of the system being studied, systems thinking works by expanding its view to take into account larger and larger numbers of interactions as an issue is being studied [9].

This thesis proposes to use Agent-based Modeling and Simulation (ABMS) methodology to study the dynamics of loyalty in market systems from the service-dominant logic perspective. Since ABMS adopts systems thinking approach to model and simulate complex adaptive systems computationally, combination of ABMS and service-dominant logic seems to have the right potential to initiate a new direction for the research on customer loyalty. In ABMS, Agents resemble actors of the real social system being modeled and are software components programmed to mimic the behaviors of their real counterparts when put together. 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 [8].

1.5. Objectives

However, building an agent-based model based on the fundamental propositions of service-dominant logic is a challenging task. Service-dominant logic proposes a completely different mindset from that of the well-established goods-dominant logic; hence this research requires a start from the scratch. In other words, due to the lack (if not unavailable) of benchmark models, building an agent-based model on the foundations of service-dominant logic requires careful attention on each of the fundamental propositions to formalize the entities of agents, define their structures, model their interactive behaviors, etc. For example, value co-creation is still a vague and abstract concept but central to this study and hence, modeling the value co-creation process among agents requires an in depth analysis. Therefore, this thesis is developed having the building of agent-based model from the service-dominant logic as a key objective. Following is the list of objectives of this thesis in the quest for answers to the research questions.

- Develop an Agent-based Model from the SD Logic perspective
- Study how 'Loyalty' change dynamically in a competitive environment
- Study the relationship between the dynamics of loyalty and the decision to switch

1.6. Thesis Contents

This thesis contains six chapters in the main text followed by few appendices to fulfill certain gaps. The chapter two discusses the important literature relevant to this research. Chapter three, which is the longest chapter, contains the details of the model, organized according to the ODD protocol. Chapter four presents the results of the basic experiments conducted using simulation. Chapter five discusses further about the model, its usefulness and applicability. It discusses the applicability of the model to the tourism industry of Sri Lanka. Chapter six provides the concluding remarks, limitations and future work.

2. Literature Review

This section of the thesis contains a review of the related literature of this research. Due to the interdisciplinary nature of this research, as illustrated by Figure 2.1, this section demands a thorough review of all major research streams related to the study. Therefore, this section is organized under three subsections namely, “Markets and Systems”, “Customer Satisfaction, Engagement and Loyalty” and “Agent-based Modeling and Artificial Markets”.

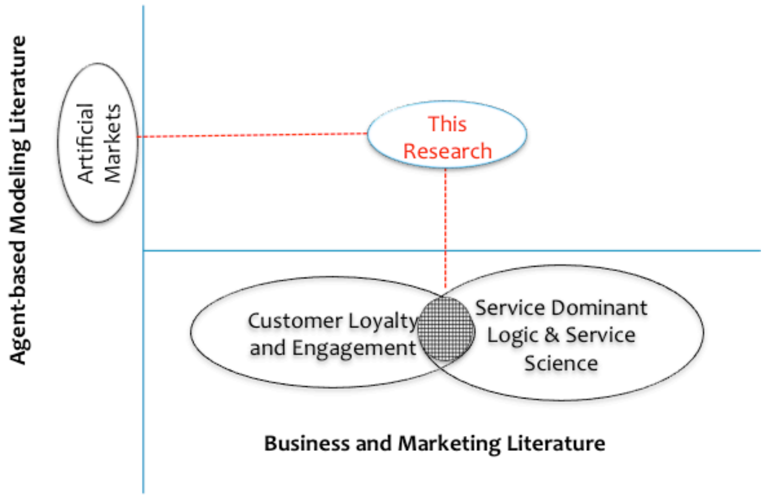

Figure 2.1: Overview of related work

2.1. Markets as Systems

The identity of Marketing is continuously evolving over a century [10]. According to the latest definition of marketing of the American Marketing Association (AMA), marketing is the activity, set of institutions, and processes for creating, communicating, delivering, and exchanging offerings that have value for customers, clients, partners, and society at large [11]. Compared to the historical stages of marketing philosophy such as “To Market” and “Market To”, modern marketing has the essence of “Marketing With (Customers)”, where customers are recognized as endogenous and as partners in the co-creation of value [10]. This indicates a shift in the emphasis of marketing research towards the systems thinking approach.

2.1.1. Service-Dominant Logic

According to Barile et al., systems view within marketing is not new although only through the service-dominant logic (SD logic) [12] and service science (SS) can one truly appreciate the full integration of systems thinking within marketing research [13]. Service-dominant logic was first introduced in 2004 and went through a refinement in 2006 [4] to be what it is known as service-dominant logic today [10]. S-D logic adopts the systems approach to the study of markets by defining markets as systems of resource integrating actors who interact by exchanging services and co-creating value [7]. The difference between Service-Dominant logic and what is called the traditional Goods-Dominant logic involves a philosophical discussion on value in the foundation of economics [14]. According to Vargo et al. [14], the traditional Goods- Dominant logic focuses on the value-in-exchange where as the Service-Dominant logic focuses on the value-in-use. Thus, the firms that believe in Goods-Dominant logic would focus on producing goods (or its intangible counterpart - services) in surplus with embedded value and distributing that surplus to maximize profits through economies of scale. In contrast, firms that adopt a Service-Dominant logic mindset would focus on increasing the adaptability, survivability and system wellbeing through competitive value propositions that primarily involve applied operant resources (i.e. knowledge and skills) and support realizing value in use. Table 2.1 presents the foundational premises of Service-Dominant logic and Table 2.2 presents a comparison of Service-Dominant logic and Goods-Dominant Logic in terms of value creation.

According to service-dominant logic, a service is an exchange of resources either in tangible or intangible form [4]. In other words, any tangible or intangible offering in the market is a means of offering a service by one actor to another, which is called a *value proposition*. A value proposition of a firm to its customers thus indicates an application of its resources, to a particular need. Therefore, a value proposition could be characterized by a set of value creating attributes, along which the firm's resources are applied and organized [15]. According to Lusch et al., S-D logic emphasis on the operant resources, which are mainly the knowledge and competence of the firm [4]. Typically, operant resources are applied on operand resources, i.e. tangible resources, to offer a service. The extent to which the resources need to be allocated along each attribute of the value proposition is a strategic decision of the firm. According to Karpen et al., the capabilities that facilitate and enhance value co-creation processes are strategic capabilities central to an organization's competitive advantage [16]. Therefore, a value proposition of a particular firm is a state that reflects its strategy in supporting value co-creation. The best combination of resources mobilized for a particular situation is also called "density" in the literature [15] [17].

Table 2.1: Fundamental Premises of Service-Dominant Logic

Foundational Premise ID	Description
FP1	Service is the fundamental basis of exchange
FP2	Indirect exchange masks the fundamental basis of exchange
FP3	Goods are a distribution mechanism for service provision
FP4	Operant resources are the fundamental source of competitive advantage
FP5	All economies are service economies

FP6	The customer is always a co-creator of value
FP7	The enterprise cannot deliver value, but only offer value proposition
FP8	A service centered view is inherently customer oriented and relational
FP9	All social and economic actors are resource integrators
FP10	Value is always uniquely and phenomenologically determined by the beneficiary

Table 2.2: Comparison of Goods-Dominant Logic and Service-Dominant Logic on Value Creation

Comparison Factor	Goods-Dominant Logic	Service-Dominant Logic
Value driver	Value-in-exchange	Value-in-use or value-in- context
Creator of value	Firm, often with input from firms in a supply chain	Firm, network partners and customers
Process of value creation	Firms embed value in ”goods” or ”services”, value is ’added’ by enhancing or increasing attributes	Firms propose value through market offerings, customers continue value-creation process through use
Purpose of value	Increase wealth for the firm	Increase adaptability, survivability, and system wellbeing through service (applied knowledge and skills) of others
Measurement of value	The amount of nominal value, price received in exchange	The adaptability and the survivability of the beneficiary system

Resources used	Primarily the operand re-sources (i.e. tangible re-sources)	Primarily operant resources (i.e. intangible resources such as knowledge and skills), sometimes transferred by embedding them in operand resources-goods
Role of firm	Produce and distribute value	Propose and co-create value, provide service
Role of goods	Units of output, operand resources that are embedded with value	Vehicle for operant re- sources, enables access to benefits of firm competences
Role of customers	To 'use-up' or 'destroy' value created by the firm	Co-create value through the integration of firm provided resources with other private and public resources

Value is defined as a change that people prefer [18]. Traditionally, value was considered as added in a tangible or intangible product as can be exchanged for something (commonly money). However, this product-based thinking ends the manufacturer's responsibility for value creation upon transfer of ownership with an exchange [15]. This is known as exchange value or value-in-exchange, where the customer becomes a destroyer of value created by the firm. In contrast, S-D logic talks about value-in-use [4], where it is argued that value cannot be added to a service upfront but has to be co-created by the beneficiary at the time of use [19]. More precisely, in a service interaction, the service provider makes a service offer through a value proposition using its resources and, the customer (beneficiary), upon acceptance of the offer, co-creates value with the help of resources possessed by him or her. The concept of value co-creation rejects the separation of the traditional value chain and proposes a value system where producer and customer in a relational system create value through the integration of their resources [15].

A useful and operational definition for value co-creation is present in the work of Novani, Kijima and Ng [20][21][22]. According to Kijima [21], value co-creation is defined as a process with four phases namely co-experience, co-definition, co-elevation and co-development. Co-experience is the sharing of internal models of preferences, capabilities and expectations of each other in a collaborative value co-creation process between customers and providers. Co-definition results in a shared common internal model by mutually learning each other's preferences, capabilities and expectations. Co-elevation is a zigzag shaped spiral up process of growing expectation of the customers and abilities of the providers and co-development pays attention to co-innovation generated by simultaneous collaboration among various entities. Co-development of service innovation is usually carried out in the context that customers evaluate and assesses the value, while providers learn from the responses from the customers. Moreover, co-experience and co-definition are short-term processes whereas co-elevation and co-development are long-term processes.

Service-Dominant logic is regarded as the provider of the right perspective, vocabulary and assumptions for modern service research [23] and the mindset of S-D logic has been acknowledged as having a staggering potential to continue to be a catalyst for important research in the field of services [24].

2.1.2. Service Science

According to Vargo et al., the scholarly activity with the most potential for development of a systems approach to understanding the market and application for marketing is that associated with IBM's effort to create a science of service – Service Science [7]. Service science is extended to incorporate management and engineering disciplines as Service Science, Management and Engineering [25]. Aiming to create the basis for systematic service innovation, service science would combine organization and human understanding with business and technological understanding to (1) explain the origins and growth of service systems; (2) solve fundamental problems such as how to invest optimally to improve service productivity and quality; and (3) produce unique service professionals and service scientists [26] [27].

Service science is the study of service systems, which are dynamic value co-creation configurations of resources (people, technology, organizations, and shared information) [27]. This abstract notion of service system [19] enables defining actors of service markets based on S-D logic. In other words, a market could be viewed as a population of interacting service systems of different kinds. According to Maglio et al., anything ranging from individuals, firms and agencies to worlds and planets could be a service system [23]. A service system is characterized by a value proposition, which helps it to agglomerate its resources in different dimensions and interact with other service systems by exchanging resources [19]. Hence, a market comprising firms (service providers) and their customers could be viewed as a platform, on which firm service systems interact with customer service systems co-creating value.

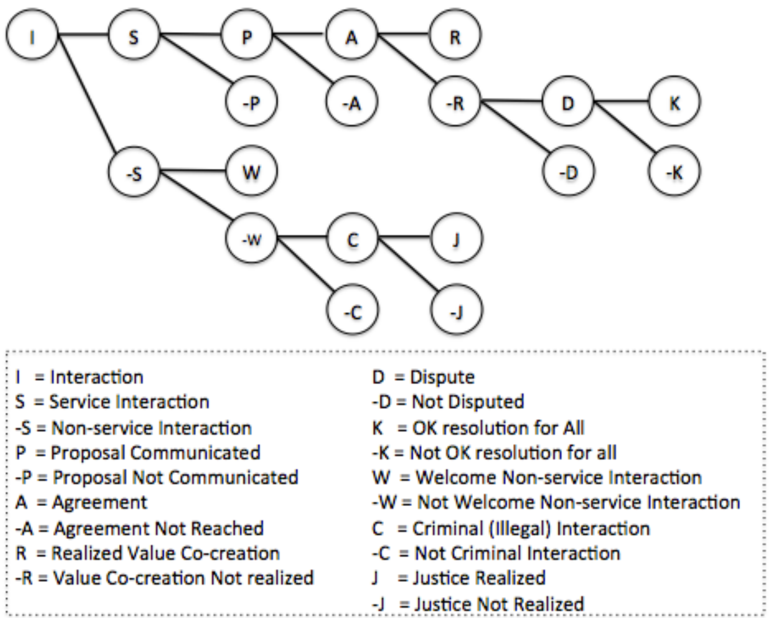

Figure 2.2: The ISPAR Model of Service System Interactions

The process of interaction between two service systems has been presented as a model of ten possible outcomes in the ISPAR (Interact-Serve-Propose-Agree-Realize) model [19]. In the ISPAR model, an interaction can be either a service interaction or a non-service interaction. Service interactions are value co-creation interactions between service providers and customers, where each service system engages in one or more of three main activities: (1) proposing a value co-creation interaction to another service system (proposal), (2) agreeing to a proposal (agreement), (3) realizing the proposal (realization). A non-service interaction involves little or no value co-creation but may act as a determinant of a future service interaction. Furthermore a non-service interaction between two service systems is voluntary and usually welcomed. An example of a welcoming behavior could be exchanging pleasantries on the street whereas an unwelcoming behavior could, for example, be committing a crime. Figure 2.2 is an illustration of the ISPAR model.

According to Maglio, one key challenge in developing a new science of service is in finding appropriate methods for modeling service systems [7]. The IBM, IfM report in 2008 suggests that, perhaps more than any other subjects, advancement in Service Science depends on models and simulations of alternative service systems designs. When data are not readily available, service practitioners need simulation tools to support their decision-making processes [28]. Based on this idea, Kieliszewski et al. discuss a modeling approach for value constellations to understand complex service system interactions [29].

2.2. Customer Satisfaction, Engagement and Loyalty

The link between customer satisfaction and loyalty is accepted widely as well as questioned equally [3][30]. As mentioned in Mittal et al., customer satisfaction has a widespread recognition over couple of decades as the key determinant of customer retention [30]. According to Oliver, loyalty refers to the extent to which customers feel committed to suppliers and do not actively seek out replacement suppliers [2]. Therefore, in the recent past, customer satisfaction management had become a strategic imperative for most firms [31] and, satisfaction has, for some, become the ubiquitous mantra for corporate success [32].

2.2.1. Measures of Customer Satisfaction

Traditional approach of measuring customer satisfaction involves the confirmation-disconfirmation of expectations [33]. As quoted by Bowden, this approach conceptualizes satisfaction as post consumption, cognitive process [3]. Those who criticize the role of measures of customer satisfaction in determining loyalty mainly claim its inability to measure the depth of customers' responses to consumption situations [34] as well as its failure to discriminate between true brand loyalty and inertia repeat purchasing [35].

According to Zeithaml et al., customers assess service performance based on two standards: what they desire and what they deem acceptable [36]. They define two levels in the expected service as the desired service level and the adequate service level. The difference between the two levels is called the "zone of tolerance". This zone of tolerance expands and contracts based on external factors.

Factor or attribute based approaches to measuring customer satisfaction supplements the confirmation-disconfirmation approach [3]. According to Allege et al., it is well established in tourism literature that both overall tourist satisfaction and a tourist's intention to return are particularly determined by his/her assessment of the destination's different attributes [37]. There exist different classifications of these attributes depending on the ways their performance contributes to the overall customer satisfaction [37]. According to Vargo et al., the primary distinction among these antecedents is that (1) some increase satisfaction when present but do not increase dissatisfaction when absent, (2) some increase dissatisfaction when absent but do not increase satisfaction when present, (3) some impact both satisfaction and dissatisfaction and negative evaluations to the extent that they are present or absent, and (4) some have no impact on satisfaction and dissatisfaction [38]. This is similar to the classification of Matzler et al. [39] following the work of Kano [40], where each attribute could be a "basic", "excitement" or a "performance" factor. Basic factors lead to extreme dissatisfaction if they do not meet expectations, yet do not increase satisfaction if they are met. Excitement factors increase satisfaction when offered but do not increase dissatisfaction when not offered. Performance factors work in both directions, generating satisfaction when work well and generating dissatisfaction when they do not.

The limitations of purely post consumption, cognitive processes to evaluate customer satisfaction have directed the research towards "affect", based on the consumers' emotional consumption responses such as customer delight [41]. However, as mentioned by Bowden [3], delight has been criticized for certain adverse effects such as increasing expectations of customers [5] and leading to habituation with regard to delighting service delivery [42].

2.2.2. Role of Customer Engagement in Dynamics of Loyalty

According to Brodie et al., within the academic marketing and service literature, very few articles used the terms "consumer engagement", "customer engagement" and/or "brand engagement" prior to 2005 [6]. Recognizing the need for a systematic scholarly inquiry into the concept of "engagement" and its conceptual distinctiveness from the other, associated relational concepts, they explore the theoretical foundations of "customer engagement – (CE)" by drawing on relationship marketing theory and service-dominant logic. The analysis derives five fundamental propositions, which are used to develop a general definition of customer engagement.

Table 2.3: The Five Fundamental Propositions of Customer Engagement

ID	Fundamental Proposition
FP1	CE reflects a psychological state, which occurs by virtue of *interactive customer experiences* with a *focal agent/object* within specific service relationships
FP2	CE states occur within a *dynamic, iterative process* of service relationships that co-creates value

FP3	CE plays a central role within a nomological network of service relationships
FP4	CE is a *multidimensional concept* subject to a context- and/or stakeholder-specific expression of relevant cognitive, emotional and behavioral dimensions
FP5	CE occurs within a specific set of situational conditions generating differing CE levels.

The five fundamental propositions of customer engagement are mentioned in Table 2.3. According to the general definition attained from those five fundamental propositions,

> CE is a *psychological state* that occurs by virtue of *interactive, co-creative customer experiences* with a *focal agent/object* (e.g. a brand) in focal service relationships. It occurs under a specific set of context-dependent conditions generating differing CE levels; and exists as a *dynamic, iterative process* within service relationships that *co-create value*. CE plays a *central role* in a nomological network governing service relationships in which other relational concepts (e.g. involvement, loyalty) are antecedents and/or consequences in iterative CE processes. It is a *multidimensional concept* subject to a context- and/or stakeholder-specific expression of relevant cognitive, emotional, and/or behavioral dimensions.

Brodie et al. further discuss several customer engagement research implications pertaining to the five fundamental propositions of CE mentioned in Table 2.3 [6].

Bowden presents a conceptual model of the customer engagement process, which determines loyalty through affective commitment towards a service provider iteratively [3]. The model, depicted by Figure 2.3, clearly differentiates the new and repeat customers. A new customer usually possesses an ill developed knowledge structure about a service provider compared to a repeat customer who has a rather developed knowledge structure with previous experience. A new customer tends to evaluate different attributes of a service (from a particular service provider) when evaluating a consumption experience, which determines his or her satisfaction and intention to return. Hence, calculative commitment is the extent to which a new customer evaluates the attribute level outcomes of a service. A positive overall evaluation of attribute level outcomes causes customer delight, which would help originating an affective commitment in the new customer. Experience of a new customer with a service provides a feedback, which enhances the knowledge structure of that customer about the particular service provider. A repeat customer on the other hand has a well-developed knowledge structure about the service of a particular service provider. Hence his or her satisfaction is assumed. The satisfaction of a repeat customer of a service provider helps developing trust on that service provider. The trust helps developing an emotional bond between the repeat customer and the service provider strengthening the affective commitment and involvement with the particular service provider. The affective commitment strengthens the loyalty of the repeat customer with the service provider, while giving feedback to improve his or her knowledge structure about the service of that particular service provider.

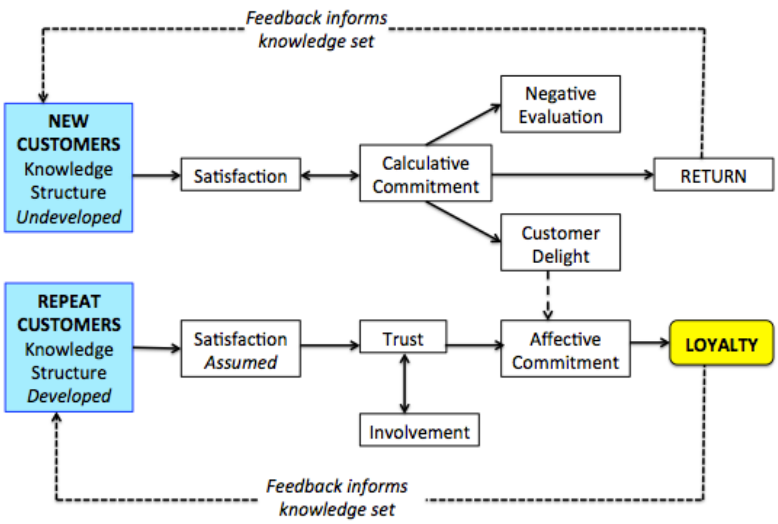

Figure 2.3: The Customer Engagement Process Model by Bowden

2.3. Agent-based Modeling and Artificial Markets

2.3.1. The Discipline of Agent-based Modeling

Agent-based modeling consist of a number of interacting autonomous agents who are represented as computerized independent entities capable of acting locally in response to stimuli or to communication from other agents [43]. These agents, when put together, interact with each other according to their local information and behavioral rules, resulting various complex patterns. In other words, these agents act as parts of a complex whole, of which the properties can only be studied by letting the parts to interact with each other. Due to this reason, agent-based modeling has become a prominent technology in studying complex adaptive systems [44].

According to Allan Kay, the best way to predict the future is to invent it [44]. In social science, this is called the generative approach, in which a generativist looking forward to explain the emergence of macroscopic societal regularities, such as norms or price equilibrium, would like to know how the decentralized local interactions of heterogeneous autonomous agents could generate the given regularity [45]. Generally, the interdependency, emergence and non-linearity inherent in the underlying processes make it difficult for humans, unassisted by computer simulations, to effectively reason about the consequences of actions in a complex system [46]. Agent-based modeling enables to generate that future, i. e. the would be world [47], in the form of a computer simulation in which a group of heterogeneous, autonomous, bounded rational agents interact locally in a explicit space [45]. The creation of silicon surrogates of real- world complex systems allows us to perform controlled repeatable experiments on the real McCoy [48].

Even though agent-based modeling has been used extensively in various domains of complex adaptive systems, there is not much evidence in the literature about using agent-based approach to study complex service systems. However, the emerging literature in complex service systems implies that there is a need for adopting such computational techniques for the progress of the discipline [13], [14], [49], [50]. According to [51], one key challenge in developing a new science of services is in finding appropriate methods for modeling service systems. Meanwhile, [52] sees that agent-based modeling techniques, first developed for artificial intelligence, are now being applied in new areas such as computational organization theory and agent-based computational economics, indicating an interdisciplinary academic shift with a potential for services. Furthermore, [53] proposes agent-based modeling as a research methodology for this emerging field, which lacks research methodologies.

2.3.2. Artificial Market Research

Artificial markets has been a popular and emerging form of agent-based social simulation, in which agents represent consumers, firms or industries interacting under simulated market conditions [54]. According to Zenobia et al. [54], there are several promising applications of artificial markets such as forecasting future market behavior, exploring market dynamics, conducting massively parallel market analysis, gaming organizational strategies for volatile new markets, and profiling products and services which do not currently exist, but which markets are poised and ready to accept. Furthermore, the recent proliferation of social networks has boosted the interest of studying the diffusion of innovations through agent-based modeling. For example, Lee et al. [55] studies pricing and timing strategies such as time to market and time to discount of a new product using agent-based simulations of behavioral consumers. Consumer agents of that model make purchase decisions for a new product referring to the characteristics of the current product they use and to the recommendations of the peers in their social network. Moreover, Garifullin et al., discuss artificial market modeling patterns using Anylogic simulation environment [56]. However, Baptista et al. [57] argues although a number of agent-based models of consumer behavior have been proposed in recent years the advantages of this approach are yet to be fully grasped by the business simulation community. This statement can be related in particularly to the emerging domain of service-dominant logic as only a handful of agent-based models have been developed based on service-dominant logic.

2.3.3. Kauffman's NKCS Architecture

The NKCS model developed by Kauffman mimics the co-evolution of multiple species in a biological ecosystem [58]. This can be likened to the process, in which stakeholders of a complex service system co-evolve outcomes, both individually as well as systemically, by providing service to each other by means of their individual competence. Thus, the NKCS model provides an appropriate and interesting basis for a rational discussion on value co-creation in complex service systems.

As the name denotes, the NKCS model is based on four main parameters N, K, C and S. Even though not mentioned, there exist two other parameters namely X and A. The NKCS model defines S number of species, each represented by a genotype comprising N number of genes. K defines the degree of inter-dependence of each gene within a genotype. In other words, each gene in a given genotype depends on K number of genes of the same genotype. C defines the degree of interdependence of each gene in a given genotype with genes of another genotype. In other words, each gene in a genotype depends on C number of genes of another genotype. Parameter X denotes the number of other species in the system that a given species interacts with. Therefore, each gene of a given genotype depends on C number of genes of each of its X interacting genotypes.

Each species has an individual fitness landscape defined by its genotype as an *N*- dimensional hypercube. Each point in this landscape has a coordinate written as a string of digits with base *A*, where *A* is a positive integer. For example, if *N = 5* and *A = 2*, points in this 5-dimensional hypercube could be identified as *00000, 00001, 00010*, etc. Each point $d = d_1, d_2, \dots, d_N$ in this *N*-dimensional hypercube has an associated fitness value □ as defined by Equation 2.1.

$$f(d) = \frac{1}{N}\sum_{i=1}^{N} f_i\{d_i, [d_{i_1,}d_{i_2}, \dots, d_{i_K}], [(d_{i_{11},}d_{i_{21}}, \dots, d_{i_{C1}}), \dots, (d_{i_{1X},}d_{i_{2X}}, \dots, d_{i_{CX}})]\} \quad (2.1)$$

Here, f_i is the fitness contribution of gene d_i, at locus *i*. However, it depends on the gene d_i, as well as the other genes that d_i is depending on. With parameter K, the gene d_i depends on K other genes denoted by $[d_{i_1,}d_{i_2}, \dots, d_{i_K}]$. Moreover, with parameters C and X, the gene d_i depends on C number of genes in each of X number of other genotypes denoted by $[(d_{i_{11},}d_{i_{21}}, \dots, d_{i_{C1}}), \dots, (d_{i_{1X},}d_{i_{2X}}, \dots, d_{i_{CX}})]$. Position values $i_1, \dots, i_K$ and $[(i_{11}, i_{21}, \dots, i_{C1}), \dots, (i_{1X}, i_{2X}, \dots, i_{CX})]$ are determined randomly. The value of f_i is determined by a function, which is defined by Equation 2.2 [59].

$$f_i: \{0, \dots, A-1\}^{(1+K+CX)} \to R \qquad (2.2)$$

Here, R is drawn from a uniform distribution in the range (0, 1) to each of its $A^{(K+XC+1)}$ inputs. The different fitness values associated with each point in a given entity's landscape entails a rugged terrain for the entity (say, a representative agent) to traverse, from valleys to peaks, looking for better fitness values. However, due to the dependency imposed by parameter C, a movement of one entity may deform the position of another (possibly many) affecting its fitness. This provokes reaction from the affected entities in return and, as the process continues, all entities in the system move to positions with better fitness values. This process is identified as co-evolution of species.

The traversal of agents in their respective landscape could be threefold as there are three standard strategies namely One-mutant change, Fitter dynamics and Greedy dynamics [58].

- **One-mutant change**: the agent chooses a single new location from the set of one mutant neighbors and if the fitness of the new location is greater than the current location, the agent moves, otherwise it stays where it is
- **Fitter dynamics**: the agent chooses a new location from the set of one-mutant neighbors, but if the fitness of that location is less than the current

location, the agent tries another neighbor, continuing until either a fitter location is found, or the set of neighbors has been exhausted

- **Greedy dynamics**: the agent moves to the location with maximum fitness in the set of one-mutant neighbors, unless the fitness of that location is less than the current location

An agent-based implementation of value co-creation processes based on the NKCS model and a related discussion is available in Rajapakse et al. [60]. Combining this work with the ISPAR model of service system interactions depicted by Figure 2.2, Rajapakse et al. in [61] present a method to develop agent-based models based on the service system abstraction and the ISPAR model, to study market systems in the light of Service-Dominant logic. Furthermore, Rajapakse et al. in [61] discuss the evolution of service providers with respect to the life cycle concept by simulating a market based on service-dominant logic [62]. Moreover, Rajapakse et al., in [63] [64] discuss an artificial market model to study the dynamics of customer loyalty from the service-dominant logic perspective.

3. The Proposed Agent-based Model

This chapter of the thesis contains a detailed description of the agent-based model developed for this research. According to Richiardi et al. [65], agent-based models need to be explained adhering to a common protocol to enhance the readability and replicability. This thesis utilizes the common protocol called the ODD (Overview, Design Concepts and Detail) introduced by Grimm et al., to describe the proposed agent-based model [66]. Particularly, the updates proposed to the ODD protocol in 2010 [67] are considered in this chapter. According to the updated ODD protocol, this discussion is organized into the structure presented by Figure 3.1 [67].

Overview	Purpose
	Entities, State variables and Scales
	Process overview and Scheduling
Design Concepts	Basic Principles
	Emergence
	Adaptation
	Objectives
	Learning
	Prediction
	Sensing
	Interaction
	Stochasticity
	Collectives
	Observation
Details	Initialization
	Input data
	Submodels

Figure 3.1: Organization of Model Details According to the ODD Protocol

3.1. Overview of the Model

The model described in this chapter corresponds to a market with many providers offering a particular service to a population of customers. As depicted in Figure 3.2, there occur service interactions between customers and service providers. Moreover, customers communicate with each other through recommendations.

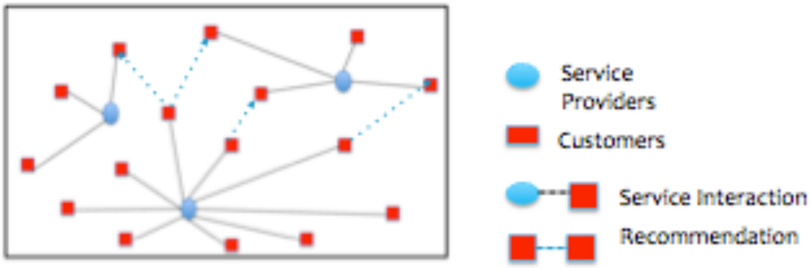

Figure 3.2: Model Overview

3.1.1. Purpose

The purpose of this model is to study the dynamic behavior of customer loyalty in a competitive market environment.

3.1.2. Entities, State Variables and Scales

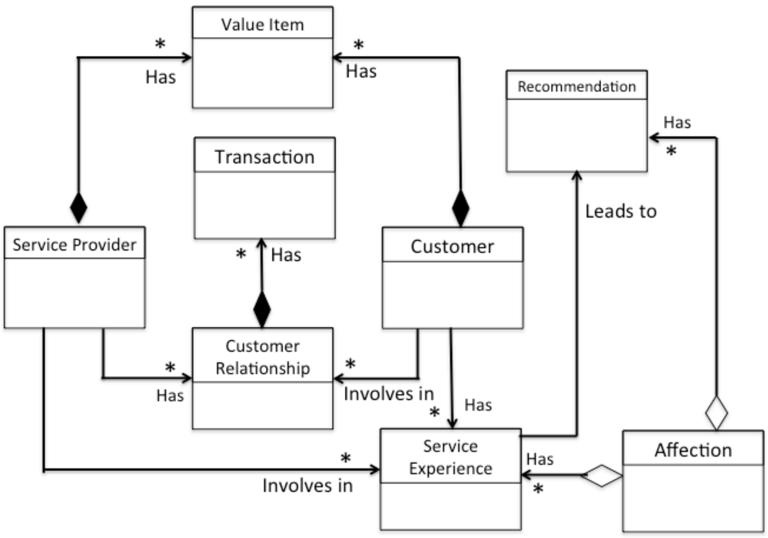

Figure 3.3: Entities and their Relationships as a Class Diagram

There are eight major entities in the system namely Value Item, Customer, Service Provider, Customer Relationship, Transaction, Service Experience, Affection and Recommendation. The class diagram in Figure 3.3 depicts the relationships among these entities. Apart from these entities, there exists a context builder to define the market conditions and a controller to act as a mediator in the actions of customer and service provider entities with their environment. Out of these entities, Customer, Service provider and Controller entities, represent the agents in the system while the rest of the entities act as data structures. Notably, the lists of methods corresponding to each entity are provided in Appendix C.

Table 3.1: Attributes of the Entity "Value Item"

Attribute	Type	Example	Description
iD	String	C1	The unique ID number of the value item
dependencyItemList	List<String>	C3,C4, P1,P2	The list of other value items with a dependency relationship
changeable	Boolean	No	Whether the state of the attribute is changeable or not
category	String	Basic	Whether this is a basic, excitement or a performance feature

Value Item: According to the *service system abstraction*, all service systems are characterized by a value proposition. A value proposition is represented in this model as a collection of *N* value creating attributes. The *Value Item* entity serves this purpose by creating instances of value creating attributes. An ID, prefixed by either "C" or "P" to distinguish between the value items of Customer and Service Provider entities respectively, uniquely identifies each value item. According to the NKC architecture, each value item depends on some other value items of the same entity as well as the opposite entity. Therefore, each value item is given a list containing the IDs of the items it has a dependency relationship as an attribute. Table 3.1 contains the attributes of this entity and their descriptions. The attribute "changeable" denotes whether the state of the value item could be changed or not. In reality, there are some items in value proposition that have a constant state. For example, a tourist can be either a local or a foreigner. The attribute "category" distinguishes whether this value item represents a basic, excitement or a performance feature according to Kano model [40].

Customer: This entity represents the “customer agents” of the system. An ID prefixed by “C” identifies a customer uniquely. Customer entity has several attributes as listed and described in Table 3.2. There, the “number of states” can take positive integers including zero and there is a corresponding state for each value-creating attribute of a given customer’s value proposition, which collectively forms the “current state” of that customer. The current state is synonymous to the “current customer profile” of that particular customer. For example, if the number of value creating attributes in the customer entity’s value proposition is 5 and the number of states is 2, a given customer’s current state could be 10010. Differences in current states distinguish customer agents from each other in terms of their customer profiles. When customer agents learn and adapt to market conditions, they dynamically change their current states by moving to neighboring states.

Table 3.2: Attributes of the Entity “Customer”

Attribute	Type	Example	Description
id	String	C001	The unique ID number of a customer
noOfStates	int	4	Total number of states available for an attribute to be at
myValueItems	List<ValueItem>	[C1,C2,…, CN]	The value items that resemble customer entity's value proposition
proValueItems	List<ValueItem>	[P1,P2, …, PN]	The value items that resemble service provider entity's value proposition
myLandscape	List<String[]>		The possible state value combinations for each attribute and respective utilities
currentState	int[]	1010101010	Current state of each attribute of the value proposition
currentExpectations	double[]	[.2,.45,.8,…]	Current expectation level on each attribute
needProbability	double	0.25	The probability of getting a service need
availableProviders	List<Provider>	[P1, P3]	List of the known providers of the customer agent
myAffectionWithProviders	List<Affection>	[A1, A2]	The customer's affection with each provider
myNeighbors	List<Customer>	[C004, C245]	List of neighbors of the customer
messageTransmitPercentage	double	0.25	The brobability of capturing a message about a new provider
currentProvider	Provider	P3	The current/lasttime service provider
currentUseContext	int	0	Current use context of the customer
grid	Grid<Object>		Current grid on which customer agents are projected
myController	Controller		The reference to the controller agent

There is also an expectation value for each value-creating attribute, which collectively form “current expectations” of the given customer. Expectation value for a given value-creating attribute is synonymous to the expected utility from that attribute during a service [36]. Initially, the expectations of individual value creating attributes are set randomly within a range of $0 - h$ ($0 < h < 1$), where h is controlled by a parameter - Customers’ adequate margin. However, customer expectations usually grow with experiences, especially with delightful experiences [42]. Therefore, expectations are set to grow by a certain quantity determined by a parameter (expectation growth rate) at each successful value co-creation.

Customer agents are stimulated by a service need, which arises based on a "need probability" at each time step and, they select a suitable service provider to fulfill that need from a known list of "available providers" where the choice often depends on the "affection with providers". When there are no known service providers to fulfill a service need, customers seek for information from their neighbors (myNeighbors). Ideally, there can be several use contexts for a given service and "my current use context" determines the use context at a given time. However the current model is assumed to have only one use context.

Table 3.3: Attributes of the Entity "Provider"

Attribute	Type	Example	Description
id	String	C001	The unique ID number of a provider
noOfStates	int	4	Total number of states available for an attribute to be at
myValueItems	List<ValueItem>	[C1,C2,..., CN]	The value items that resemble provider entity's value proposition
proValueItems	List<ValueItem>	[P1,P2, ..., PN]	The value items that resemble service customer entity's value proposition
myLandscape	List<String[]>		The possible state value combinations for each attribute and respective utilities
currentState	int[]	1010101010	Current state of each attribute of the value proposition
currentExpectations	double[]	[.2,.45,.8,...]	Current expectation level on each attribute
myCheckAttributes	int[]	[2,5,6]	The positions of attributes to be checked when accepting a service request
myCurrentContext	int	0	The current context of the provider
transactionNumber	int	100	An incremental value based on transactions with customers
tansactionLog	List<double>	125.00	A list to determine the number of transactions at each time step
myAverageAffection	double	0.15	The average affection available in the system for this provider
visitedStates	List<String>	[100011, 1001]	The list of previously visited states - (to avoid revisits)
grid	Grid<Object>		Current grid on which customer agents are projected
myController	Controller		The reference to the controller agent

Provider: This entity represents the "service provider agents" of the system. It is assumed that all providers of this entity offer a single service to the customers. Similar to customer agents, there are unique IDs prefixed by "P" to identify each service provider in the system. The attributes of the provider entity are listed and described in Table 3.3. Objectives and definitions of most attributes of the provider entity, especially the current state and the expectations, are similar to the customer entity. The check attributes (myCheckAttributes) represents a subset of value items of customer entity's value proposition. When a provider agent gets a service request from a customer agent, the provider agent uses the potential utility from this subset to accept or reject the request. For example, when a tourist seeks for accommodation at a hotel, the hotel usually asks for the credit card as a security. If the tourist cannot produce a valid credit card, the service request will be rejected. Once a service provider agrees to service a customer, a transaction occurs and the provider updates all transactions in a log.

Service Experience: Service experience corresponds to an outcome of a service interaction between a customer and a service provider. In the model, customer agents keep track of their service experiences with each service provider. Each service experience contains the IDs and Current States of both parties involved as well as the response of the provider, delight from the service and the contribution of the service experience to the overall trust. Table 3.4 contains a description of all attributes of this entity.

Table 3.4: Attributes of the Entity "Service Experience"

Attribute	Type	Example	Description
myID	String	C250	ID of the customer agent
providerID	String	P3	ID of the provider with whom the service interaction took place
myState	int[]	[1100110011]	State of the customer agent
providerState	String	[1010101010]	State of the provider agent
wasAccepted	boolean	TRUE	Whether the service request accpeted or rejected
delight	double	0.35	Delight of the service interaction, if any
trustContribution	double	0.45	Corresponding trust value from the exponential function

Recommendation: Recommendation corresponds to a peer recommendation received by a customer about an experience with a particular service provider. A recommendation contains the ID, current state and current expectations of the recommending peer as well as the ID of the provider being recommended. Table 3.5 contains a description of the attributes of this entity.

Table 3.5: Attributes of the Entity "Recommendation"

Attribute	Type	Example	Description
RecommenderID	String	C245	ID of the recommending peer
ProviderID	String	P2	ID of the provider being recommended
RecommenderCurrentState	int[]	[1001001001]	Current state of the recommending peer
RecommenderExpectations	double[]	[.5, .44,…,.55]	Current expectations on each attribute of the recommending peer

Table 3.6: Attributes of the Entity "Affection"

Attribute	Type	Example	Description
providerID	String	P1	ID of the provider considered
proCurrentState	int[]	[1001001001]	Current State of the provider considered
myCurrentState	int[]	[1110001110]	Current state of this customer agent
experience	List<ServiceExp>	[SE1,SE2,…]	Service experiences with this provider so far
recommendations	List<Recomm.>	[R1, R2,…]	Recommendations received about this provider so far
trust	double	0.44	Trust for this provider so far
RecommendationStrength	double	0.32	Strength of peer recommendations accumulated so far
isInvolved	boolean	TRUE	Whether the customer is involved with the provider or not

Affection: The entity of affection has attributes to keep data about a customer's affection towards a particular service provider. Apart from the general details such as ID and current states, it contains a full history of service experiences and peer recommendation with regard to a particular provider. It also contains the overall "trust" and "recommendation strength" with regard to that provider, which are used to calculate the "affective commitment" of a customer towards a given provider. Apart from these, the attribute "is involved" determines whether the customer agent is involved with the respective service provider based on a "trust threshold". Table 3.6 contains a description of the attributes of this entity.

Table 3.7: Attributes of the Entity "Transaction"

Attribute	Type	Example	Description
transactionID	String	125.0-100	Transaction is identified by a unique ID
customerID	String	C234	ID of the customer involved in the transaction
customerState	int[]	[1110001110]	State of the customer interacted for the transaction
feedback	double	0.5	Feedback on the customer delight

Transaction: Transaction refers to service interactions accepted by the service providers. In this model, provider agents keep track of all of their transactions with customers. A transaction contains a unique transaction ID, customer ID, customer's state and the customer's feedback as described in Table 3.7.

Table 3.8: Attributes of the Entity "Customer Relationship"

Attribute	Type	Example	Description
cusID	String	C111	ID of the customer considered
transactionHistory	List<transaction>	[T1, T2, ..]	History of all transaction with this customer
myUtility	double	0.55	Utility to work with the particular customer

Customer Relationship: As the name denotes, the customer relationship entity corresponds to the individual relationships between providers and customers. It contains attributes to keep customer ID, a history of all transactions with that customer and the provider's overall utility with that customer as described in Table 3.8.

Table 3.9: Attributes of the Entity "Context Builder"

Attribute	Type	Example	Description
customerN	int	10	Length of customer entity's value proposition
providerN	int	10	Length of provider entity's value proposition
K	int	2	No. of attributes of the same entity a given value item depends on
C	int	2	No, of attributes of the opposite entity a given value item depends on
noOfStates	int	4	No. of states in the system
noOfContexts	int	1	No. of use contexts
cusChngblAttPercentage	double	0.2	Changeable attribute percentage of the customer's value proposition
proChngblAttPercentage	double	0.2	Changeable attribute percentage of the provider's value proposition
noOfCustomers	int	1000	No. of customers in the system
noOfProviders	int	2	No. of providers in the system
customerValueItemList	List<ValueItem>	[C1,C2,...]	List of value items in the customer entity's value proposition
providerValueItemList	List<ValueItem>	[P1, P2, .]	List of value items in the provider entity's value proposition
context	context<Object>		context of the model defined as an object in Repast

Context Builder: Context Builder is a default class in Repast Symphony [68] that builds up the context of the model. Description of this class is necessary since this model is implemented using Repast Symphony as the agent-development platform. Attributes defined in context builder correspond to some of the "parameters of the model" described in a subsequent sub subsection. Table 3.9 contains a list of all attributes along with their descriptions.

Controller: A single agent represents the controller entity in the system. The controller agent's attributes correspond to most of the "parameters of the model" described in the next sub subsection. The controller agent acts as a mediator in some of the actions of customers and service providers and also does the reporting of the simulation results as MS Excel files. Table 3.10 contains a list of all attributes along with their descriptions.

Table 3.10: Attributes of the Entity "Controller"

Attribute	Type	Example	Description
customerN	int	10	Length of customer entity's value proposition
providerN	int	10	Length of provider entity's value proposition
K	int	2	No. of attributes of the same entity a given value item depends on
C	int	2	No, of attributes of the opposite entity a given value item depends on
noOfStates	int	4	No. of states in the system
noOfUseContexts	int	1	No. of use contexts
noOfCustomers	int	1000	No. of customers in the system
noOfProviders	int	2	No. of providers in the system
myContext	context<Object>		context of the model defined as an object in Repast
cusAdequateMargin	double	0.4	Upper value of the customers' initial expectations
proAdequateMargin	double	0.4	Upper value of the providers' initial expectations
cusLandscape	List<String[]>		State value combinations and respective utilities of each value item
proLandscape	List<String[]>		State value combinations and respective utilities of each value item
customerValueItemList	List<ValueItem>	[C1,C2,...]	List of value items in the customer entity's value proposition
providerValueItemList	List<ValueItem>	[P1, P2, .]	List of value items in the provider entity's value proposition
avlCustomers	List<Customer>	[C1, C2..]	List of all available customers
avlProviders	List<Provider>	[P1, P2, ..]	List of all available providers
conditionsChecked	int[]	[1,3,7]	Positions of the attributes that will be checked by providers at a service request
addCompetition	boolean	TRUE	Whether to add competition during the simulation run
addIntervals	List<Integer>	[100, 500, 800]	When to add competition if above is TRUE
Output file streams			Many output file streams in .xls format have been defined

Model Parameters: Table 3.11 contains the key parameters of the model, most of which correspond to state variables of previously explained entities. In this model, a value proposition is a collection of N value creating attributes. Therefore, the "customerN" and "providerN" correspond to the sizes of the value propositions of customer and service provider entities respectively. Parameters K and C correspond to the dependency structure imposed on the value items of Customer and Provider entities to mimic co-creation of value. As mentioned in sub subsection "Value Item", states of some value items of both value propositions are set as not changeable. The two parameters "proChngblAttPercentage" and "cusChngblAttPercentage" denote what percentages of attributes on each entity's value proposition are set as changeable. The parameter "Loyalty Biasness" lets agents a percentage of freedom to make a random decision when selecting a service provider. As the parameter is set to 0.75 in the default case, there is a 25% chance for an agent to make a random selection without considering the loyalty. Both parameters "Trust of Conformity" and "Loss of Unconformity" deals with the trust gained from satisfaction. If the satisfaction is positive, i.e. > 0, the quantity of "Trust of Conformity" will be added to the overall trust to represent the effect of "delight". Similarly, if the satisfaction is negative, the quantity of "Loss of Unconformity" will be reduced from the overall trust to represent the effect of dissatisfaction. Since a loss affects more according to the Prospect Theory [69], the value of "Loss of Unconformity" is set larger than "Trust of Conformity". "Innovation Frequency" determines how often the service provider agents consider changing their current state to a better state, i.e. a state that makes both the provider agent and its customers better off. This is done in conversation with the top customers of the respective provider agent determined by the total number of transactions done with that provider. The size of the sample of top customers is determined by the parameter "Innovation Sample Size". Parameters of Lambda and Beta correspond to the determination of "Affective Commitment" explained in sub subsection 3.2.1 (Basic Principles)

Table 3.11: Model Parameters

Parameter	Type	Default Val.	Description
customerN	int	10	Length of customer entity's value proposition
providerN	int	10	Length of provider entity's value proposition
K	int	2	No. of attributes of the same entity a given value item depends on
C	int	2	No, of attributes of the opposite entity a given value item depends on
noOfStates	int	4	No. of states in the system
noOfUseContexts	int	1	No. of use contexts
noOfCustomers	int	1000	No. of customers in the system
noOfProviders	int	2	No. of providers in the system
cusAdequateMargin	double	0.4	Upper value of the customers' initial expectations
proAdequateMargin	double	0.4	Upper value of the providers' initial expectations
addCompetition	boolean	FALSE	Whether to add competition during the simulation run
cusChngblAttPercentage	double	0.8	Changeable attribute percentage of the customer's value proposition
proChngblAttPercentage	double	0.8	Changeable attribute percentage of the provider's value proposition
Need Probability	double	0.25	Probability that a customer gets a servic need at a given time step
Loyalty Biasness	double	0.75	The probability that a provider is selected based on loyalty
Trust of Conformity	double	0.25	Gain of trust when satisfaction is positive
Loss of Unconformity	double	0.4	Loss of trust when satisfaction is negative
Innovation frequency	int	10	In what intervals does a service provider conduct an innovation attempt
Innovation Sample Size	int	50	Sample size of top customers considered by providers when innovating
Message Transmit Percntage	double	0.25	Probability that a customer captures a message about a new service provider
Lambda (Gain)	int	6	Lambda value in determining the positive side of the exponential function
Lambda (Loss)	int	10	Lambda value in determining the negative side of the exponential function
Beta	long	50,000	Smoothing parameter in the calculation of recommendation strength
Expectation Growth Rate	double	0.05	Growth of expectations at each successful value co-creation experience
Trust Threshold	double	0.25	Trust threshold to determine if a customer is involved or not

Spatial Units: The customer agents are located on a grid structure, which determines the neighbors of each customer. Neighbors of a given customer agent are important for that agent to receive information about new providers, receive recommendations and to make comparisons of the service experiences.

Environment: There is only one condition determined by the environment, which is the need probability. The need probability determines how likely a given agent gets a service need at a given time step.

Temporal Units: As this is a typical abstract model, the temporal units, i.e. time steps, are not defined as days, months or years.

3.1.3. Process Overview and Scheduling

Building Model Context: The model starts with the "Context Builder" building the context of the model based on some of the model parameters. Building the context involves making the two value propositions of Customer and Provider entities, building the utility landscapes of the two entities and adding a Controller to the context with information about the value propositions and the utility landscapes built. The controller agent then creates the number of Customer and Provider agents specified in the parameters and adds them to the context. After adding the agents to the context, the controller sends a message to all customer agents about each of the available service provider in the system and, the customer agents grab the message based on the probability specified in the parameter "message transmit percentage". Details of the sub processes in this process are given in 3.3.3.

The Market Process: The market process starts with customers getting a service need. At each time step, customer agents get the need for service based on the parameter "Need Probability". Once a customer agent is aroused with a service need, the sub process "selecting a provider" depicted in Figure 3.4 begins.

As depicted by Figure 3.4, a customer agent aroused with a service need first look for any known providers. If there are known providers, the agent chooses one out of them. In this selection, there is a chance for the customer agent to choose the provider either randomly or based on the loyalty depending on the value of the parameter "Loyalty Biasness". For example, if Loyalty Biasness = 0.75, there is a 25% chance for picking up the provider randomly out of all known providers without considering the loyalty. The loyalty-based selection involves computing the loyalty with each available provider and if the process failed to identify any provider to whom the customer agent is loyal too a provider is selected randomly. On the other hand, if there are no known providers available, the customer agent queries its neighbors for information about new provider and select one randomly if new providers could be found or give up the service need otherwise. The sub process of loyalty-based selection is explained in 3.3.3.

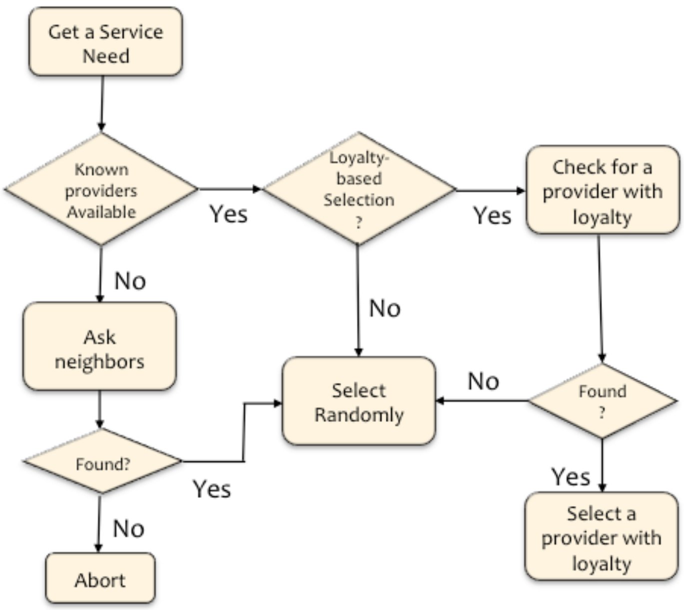

Figure 3.4: Process of Selecting a Provider at a Service Need

Once a provider is selected, the agent moves to the service interaction phase. A service interaction is a sequential process that involves actions of both the customer agent and the provider agent. This process is based on the theoretical foundations of the ISPAR model of service system interactions [19]. Figure 3.4 depicts the process of service interaction.

At the beginning of the process, the customer agent checks if it is involved with the selected provider. The involvement considered here is analogous to what has been defined as enduring involvement by Warrington et al. [70]. There it is assumed that an involved customer has a strong relationship between the product and the customer's centrally held values across all purchase situations. Based on Bowden [3], we define involvement as determined by the trust towards a provider. Thus, a customer agent determines whether it is involved or not with the selected provider based on the parameter "Trust Threshold". An involved customer agent hence seeks for better states, which are more likely to let it co-create better value with the selected provider than its current state. The better states are sought in the one-mutant neighborhood [58] of the state space.

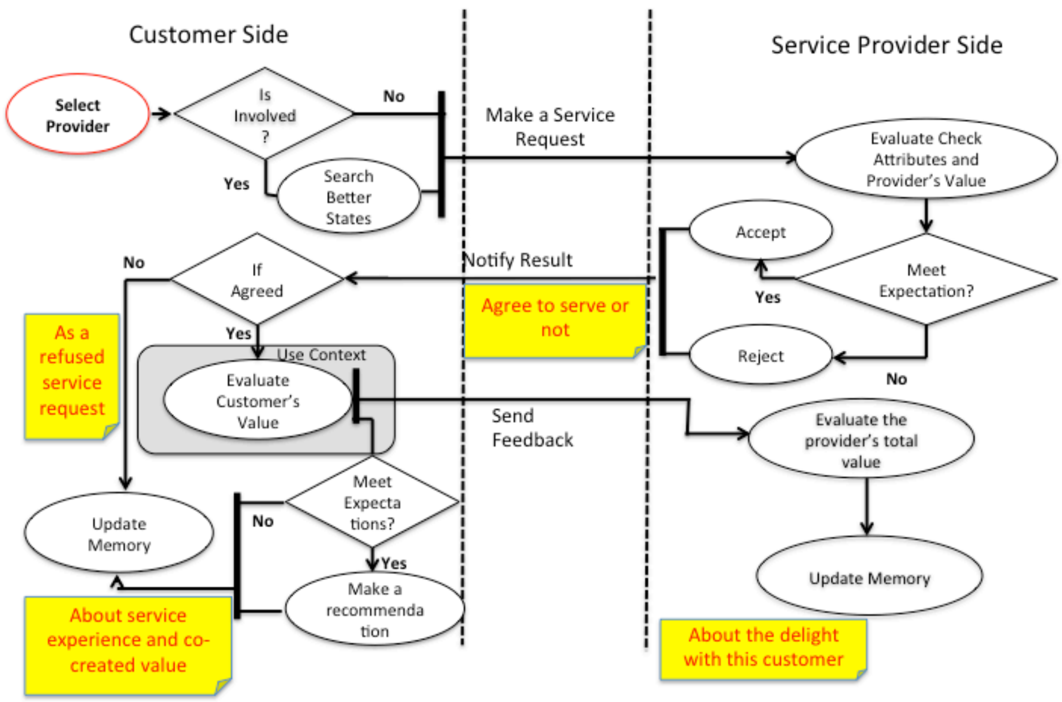

Figure 3.5: The Process of Service Interaction Between a Customer and a Provider

Whether involved or not, customer agent then make a service request to the selected provider agent. The respective provider agent then evaluates the service request based on a set of check attributes on the customer agent's value proposition. For example, possession of a valid credit card is required to reserve a hotel room in most cases, which is analogous to a check attribute. If the states of the check attributes meet the expectations of the provider agent, it accepts the service request or reject otherwise.

If the provider agrees to serve, the customer agent reaches the use context where it uses the service while co-creating value. If the perceived value exceeds or meets the expectations of the customer agent, it becomes a satisfied customer and makes a recommendation to its neighboring customer agents on the grid. The service experience, whether satisfactory or not, is updated in the memory. Furthermore, a rejection of a service request is also saved in the memory for future reference.

The customer agent further gives a feedback to the provider agent after using the service. The provider agent also evaluates the total value in the transaction and updates its memory of the relationship with the particular customer agent in terms of feedback received from the customer and the delight experienced by it. Further details of the sub models related to the service interaction process are explained in 3.3.3

3.2. Design Concepts

3.2.1. Basic Principles

This sub subsection contains the general concepts underlying the design of the model as recommended in the guidelines of the ODD protocol [65].

Value Co-creation: This model discusses mainly about service interactions between service system agents. In the case of service interactions, the concept of "Value Co-creation" is of highest significance. Therefore, the model is designed emphasizing the value co-creation of agents in their service interactions using the NKCS architecture of Kauffman [58].

In this model, a value proposition is conceptualized as a list of N value creating attributes [15]. Therefore, we define a value proposition as a list of N Value Items. The size of N for customer entity and provider entity is determined by the parameters customerN and providerN respectively. We define a state vector corresponding to the value proposition for each agent in the system, which reflects the "current state" of a particular agent with respect to the value proposition of its entity. State (d) of a given agent corresponding to a particular value-creating attribute of its value proposition is determined by the parameter D, where d ∈ D. For example, if D = 4, d ∈ {0,1,2,3}. This representation is illustrated in Figure 3.6.

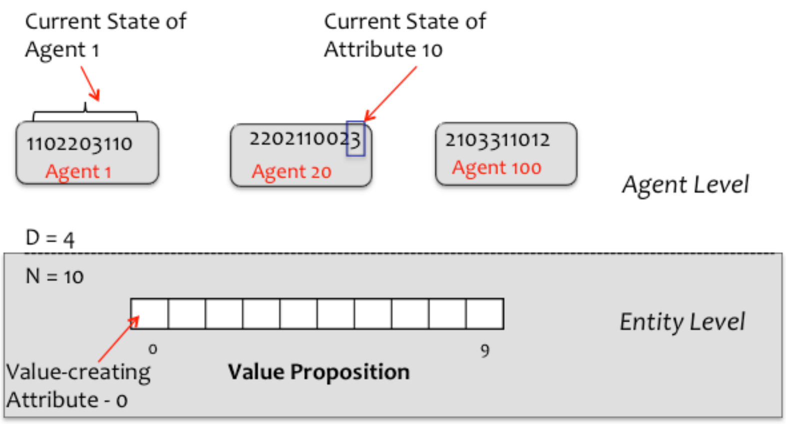

Figure 3.6: Illustration of the Representation of Value Proposition and Current States

The current state of an agent is a reflection of its profile. When it comes to providers, the current state of a provider agent reflects its current service level or the "business profile". In other words, it shows the agent's current level of operant resources used along each attribute of the value proposition to transform its operand resources into a service offer. For example, “providing Internet access to guests” could be one attribute of a hotel’s value proposition in the tourism market, in which the different ways of providing access such as “setting up Wi-Fi zones inside hotel premises”, “giving access on request at a charge” and “giving in-room Wi-Fi access to all residents” could be the states. When it comes to customers, the current state of a customer agent is a reflection of its “customer profile”. In other words, it reflects the level of operant resources of the customer along each attribute of its value proposition combined with its operand resources. For example, a tourist in the hotel industry could be “adventurous”, which would be an attribute of his or her value proposition whereas High, Moderate and Low could be possible states of that attribute.

Each attribute of the provider agent’s value proposition has a value contribution to the perceived value by the customer. On the other hand, each attribute of the customer agent’s value proposition has a contribution to the overall value perceived by the provider in the service interaction. Therefore, in a given service interaction $i, i \in I$, involving two instances, a customer - x $(x \in \bar{X})$ and a provider – y $(y \in \bar{Y})$, the perceived utility of x $(= u_i^x)$ is represented by Equation 3.1. In this representation, $\bar{X}$ denotes the entity of customers whereas $\bar{Y}$ denotes the entity of service providers.

$$u_i^x = \frac{\sum_{n=0}^{N_{\bar{Y}}-1} u_{i\,\bar{Y}}^{n}}{N_{\bar{Y}}} \qquad (3.1)$$

Here, $N_{\bar{Y}}$ denotes the number of value creating attributes of the value proposition of service provider entity and $u_{i\,\bar{Y}}^{n}$ denotes the utility contribution of the attribute *n* of service provider entity’s value proposition to the overall utility perceived by *x* in the interaction *i*. A similar equation could be written to determine the perceived value of the provider agent, *y*, in the same interaction.

Typically, the value contribution from one attribute does not entirely depend on the resource level along that attribute. It depends also on the states of few other attributes of the same entity as well as the states of some attributes of the opposite entity. For example, the perceivable value by a tourist from the attribute “Internet accessibility” at a hotel would depend not only on the type of accessibility provided (i.e. the state of the attribute itself) but also on the structure and materials used to build its rooms (internal) as well as “whether the customer possesses a laptop” and “whether he or she knows how to connect it to the network” (external). This dependency on the resource levels (i.e. states) of some attributes of the opposite entity, i.e. the value perceiving entity, is analogous to the concept of value co-creation, which implies a collaboration in terms of the contribution of resources. Value co-creation typically talks about the value realized by the beneficiary of the service - i.e. the customer at the time of use. However, this model conceptualizes a realization of value in the opposite direction of the service interaction as well. That is, the provider agent too realizes a value in the interaction depending on the match between its profile and the customer agent’s profile. For example, a budget hotel would create better value with backpackers than with mass tourists.

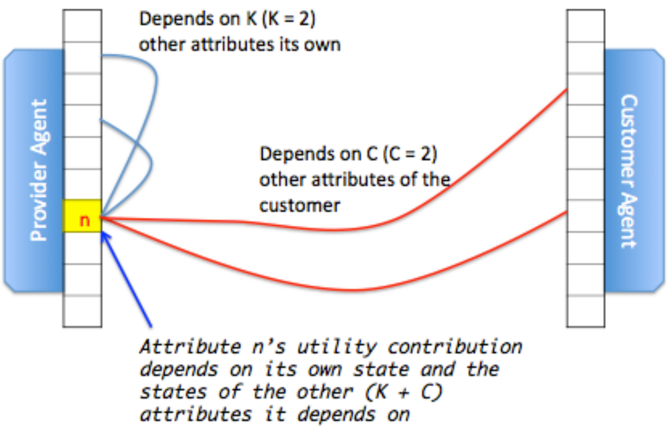

Figure 3.7: The Dependency Structure of Value Items according to the NKCS Model

This dependency of attributes is represented using Kauffman's NKCS architecture [58]. Each value item on a given value proposition is assumed to be depending on K other value items of the same value proposition as well as C other value items of the opposite entity's value proposition. Here, K and C are system parameters. Figure 3.7 depicts this dependency structure where, for example, the attribute *n* of providers' value proposition depends on K = 2 other attributes of its own and C = 2 other attributes of the customers' value proposition.

Equation 3.2 represents this dependency structure in algebraic form. According to that function, the utility contribution of any attribute *n* of the provider entity's $(\overline{Y})$ value proposition $u_{i\,\bar{Y}}^{n}$ is defined as a function of the state of that attribute $(d_{i\,\bar{Y}}^{n})$, states of *K* other attributes of its own $((d_1^n \ldots d_K^n)_{\bar{Y}})$ and states of *C* attributes of the customer entity $((d_1^n \ldots d_C^n)_{\bar{X}})$. In other words, $u_{i\,\bar{Y}}^{n}$ depends on $D^{(1+K+C)}$ number of state value combinations. Similarly, Equation 2 can be written for the attributes of the customer entity $\bar{X}$ as well.

$$u_{i\,\bar{Y}}^{n} = f(d_{i\,\bar{Y}}^{n}, (d_1^n \ldots d_K^n)_{\bar{Y}}, (d_1^n \ldots d_C^n)_{\bar{X}}) \qquad (3.2)$$

Equation 3.3 determines the utility *R* returned by function *f* for each of the $D^{(1+K+C)}$ state value combinations. Here, *R* is drawn from the uniform distribution [59].

$$(f^n) : \{0 \ldots D-1\}^{1+C+K} \rightarrow R \qquad (3.3)$$

The state value combinations related to the utilities of each attributes of a given entity's value proposition determines the utility landscape of the opposite entity. For example, the possible state value combinations for each attribute of the provider entity's value proposition and their respective utility values form the utility landscape of the customer entity.

Notably, the value proposition and the utility landscape for a given entity are common for all agents of that entity. Agents only differ from each other in terms of their current states, which denote their current resource level along each attribute of the value proposition.

Determining Loyalty: The determination of loyalty is required when a customer agent is about to select a provider based on the loyalty towards that provider. This model assumes loyalty as formed dynamically with customer's value co-creation experiences as well as peer recommendations over time.

According to Bowden, *Affective Commitment* is the key determinant of loyalty [3]. Furthermore, *Trust* is the key determinant of Affective Commitment. On the other hand, with the modern day communication technologies, *peer recommendations* do a significant impact to the customers' affection towards a particular provider or a brand. Peer recommendations can be reasonably put into the category of *Non-service Interactions* of service systems defined by Spohrer et al., which act as catalyst for future service interactions [19]. Therefore, *Affective Commitment* of customer *x* towards provider *y* $(A_x^y)$is defined as the sum of *Trust - T* and *Recommendation Strength* - $1/b\beta$ as depicted by Equation 3.4.

$$A_x^y = \frac{\sum_{m=1}^{q} T_x^y}{q} + \frac{\sum_{m=1}^{z} 1/\beta b}{z} \qquad (3.4)$$

Here, q is the total number of service interactions that have taken place between the customer x and the service provider y where as z is the total number of recommendations x has received from its peers about y.

The strength of a peer recommendation, $1/b\beta$, is defined in this model in such a way that the strength becomes inversely proportional to the distance – *b,* between the receiving customer and the recommending peer in terms of their current states. Since the current state of a customer agent is its customer profile, the distance between the profiles of two customers indicates to what extend the two customers are different. Hence, the likelihood of one customer's recommendation to give similar result to the other is higher when the distance between the two customers is lower. Here, β is a smoothing parameter, which is used to avoid the quantity *Recommendation Strength* over affecting the *Affective Commitment*.

According to Bowden, *Trust* is determined by *Satisfaction* and *Delight*. This model uses the approach of assessing the individual attributes of the value proposition against the expectations of customers to determine customer satisfaction and delight [3][37]. Therefore, satisfaction along a particular attribute on the provider entity's value proposition is defined as the difference between the Perceived Utility – PU and the Expected Utility – EU. However, following the categorization of satisfaction factors by Matzler et al., this model define each of the attributes of the provider entity's value proposition as *basic factors*, *excitement factors* and *performance factors* [39]. Therefore, the algorithm in Figure 3.8 is used to determine the satisfaction along each attribute *n* - S_n.

If (PU > EU) AND n = "Basic"
    $S_n = 0$
Else
    $S_n = PU - EU$

If (PU > EU) AND n = "Excitement"
    $S_n = PU - EU$
Else
    $S_n = 0$

If n = "Performance"
    $S_n = PU - EU$

Figure 3.8: Algorithm that Determines Satisfaction on Attributes based on Matzler et al.'s Categorization

The average of satisfactions along each attribute determines the overall satisfaction of the customer agent in the interaction *i* - $S_{x\,i}^{y}$, which is given by

Equation 3.5. There, the customer is satisfied with delight if $S_{x\ i}^{y}$ > 0, just satisfied if $S_{x\ i}^{y}$ = 0 and dissatisfied if $S_{x\ i}^{y}$ < 0.

$$S_{x\ i}^{y} = \frac{\sum_{n=0}^{N_{\bar{Y}}-1} S_n}{N_{\bar{Y}}} \quad (3.5)$$

The model refers to a sigmoid function defined based on the cumulative distribution function of exponential distribution, to determine the *trust value* corresponding to a particular value of satisfaction. Equation 3.6 contains the definition of the sigmoid function whereas Figure 3.9 depicts the corresponding graph of the function. The use of a sigmoid function to determine the trust based on satisfaction is inspired by the Prospect Theory [69].

$$f(S) = \begin{cases} -\left(1 - 1/e^{-S\lambda_{Loss}}\right) - Loss_{Unconformity}, & S < 0 \\ \left(1 - 1/e^{S\lambda_{Gain}}\right) + Gain_{Conformity}, & S \geq 0 \end{cases} \quad (3.6)$$

Here, λ_{Gain} and λ_{Loss} correspond to the input parameters Lambda (Gain) and Lambda (Loss) respectively whereas $Loss_{Unconformity}$ and $Gain_{Conformity}$ correspond to the input parameters Loss of Unconformity and Trust of Conformity respectively. In the graph of Figure 3.9, λ_{Gain} = 6, λ_{Loss} = 10, $Loss_{Unconformity}$ = 0.25 and $Gain_{Conformity}$ = 0.1. Notably, the parameters are set complying with the prospect theory, as a loss is likely to result a higher impact.

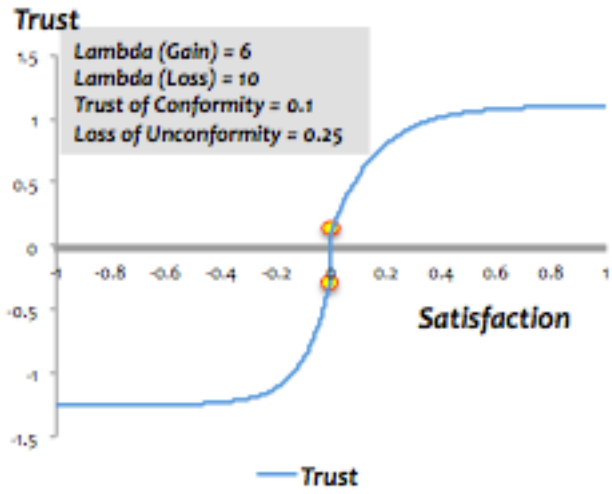

Figure 3.9: The Correspondence between Trust and Satisfaction

Once the Affective commitment is determined according to the Equation 3.4, Loyalty for a given provider is calculated as the share of affective commitment of that provider. Equation 3.7, which is based on the "multinomial logit model" [55] [71], provides this calculation. There, M is the total number of providers the customer agent x has with a positive affection. Notably, the provider agents with negative affections are ignored for this equation.

$$L_x^y = \frac{A_x^y}{\sum_{i=1}^{M} A_x^i} \qquad (3.7)$$

Here, the loyalty *L* represents a share of affection the provider *y* enjoys in customer *x*'s mind. The choice of a provider based on this quantity is probabilistic where there is a higher chance of getting selected if the loyalty share is higher.

3.2.2. Emergence

There are two main emerging patterns considered in this thesis as important. One is the dynamic pattern of the average affection of the providers in the system. Average affection of a given provider is the average of affections of all customer agents in the system towards the particular provider at a given time step. The other emerging pattern is the change of market share over time. The market share of a provider at a given time is the total number of transactions done by that particular provider at the particular time step.

3.2.3. Adaptation

Adaptation in this model mainly occurs in the population of customer agents. An involved customer with a selected provider compares its value in the previous experience with the same provider against the values received by its neighboring customers with the same provider before starting a service interaction. If a neighboring customer who creates better value with the same provider is found, the customer moves to the one-mutant neighborhood that reduces the distance between itself and the neighbor. If more than one neighbor with better value is found the agent moves to the direction of the nearest neighbor. Since the distance between neighbors determines the similarity of their profiles, moving to the direction of a nearest neighbor is analogous to changing personal traits to be alike.

3.2.4. Objective of Adaptation

The main objective of adaptation of the customer agents is improving their ability to co-create value with the providers they are involved with. Since co-creation involves the personal traits on the customers' value proposition, change of state of one trait is expected to improve the potential to co-create value with a given provider by a certain amount.

3.2.5. Learning

Customer agents learn by comparing their experiences with those of their neighbors in terms of the co-created value, which lead them to adapt. In contrast, provider agents learn new states at which their customers can co-create value better in conversation with their top customer. Each provider agent periodically contact a sample of its top customers to learn better states in their one-mutant neighborhood, which let the top customers co-create better value. It is assumed that the top customers would spread the message to others eventually. The time interval between two innovation attempts and the size of the sample of top customer is determined by the two input parameters, Innovation Frequency and Innovation Sample Size.

3.2.6. Sensing of Agents

Customer agents mainly sense a service need arouse internally. It is considered as a sensing from the environment. Provider agents sense their market share periodically and take actions to learn new states in the one-mutant neighborhood in case if a decline is observed.

3.2.7. Predictions

Customer agents mainly predict their future ability to co-create value. In that sense, they assume that trying to be like their neighbors who co-create better value with the same provider would enable them to co-create better value. However, this imitation process is not allowed to happen at once, as it is unlikely in reality that a customer could acquire all traits of another customer at once. Therefore, only an incremental learning is permitted. In the case of providers, they predict the ability of their customers to better co-create value when changing their current state to one of the neighboring states. Since it is practically impossible to talk to all customers, the providers check the potential of a decision to move with a sample of its top customers.

3.2.8. Interactions

All agent interactions in the model are direct interactions. The customer agents interact with provider agents to get a service need fulfilled. The customer agents interact with customer agents to make a recommendation. According to the ISPAR model of service system interactions, recommendations are assumed to be non-service interactions.

3.2.9. Stochasticity

Stochasticity is present in the customers' decision to select a service provider when they get a service need. The parameter "Loyalty Biasness" controls the percentage of Stochasticity permitted in the system. For example, if Loyalty Biasness = 0.75, there is a 0.25 chance for an agent to choose a provider randomly ignoring the loyalty factor.

3.2.10. Observation Data

Data collected for observation and analysis falls into the category of macro level and micro level. In the macro level of the analysis, mainly the data related to the market share, i.e. the number of transactions in each time step, and the average affection, i.e. the average affection of each service provider in each time step are collected. For the micro level analysis, data related to the individual agents' behavior is collected. For example, during the simulation run, who are the defected agents and who are the retained agent, what was the affection at the time of switch, who the newly selected provider is and what the affection of the newly selected provider is at the time of a switch. Data is collected into spreadsheets using MS Excel as file outputs.

3.3. Details

3.3.1. Initialization

The model parameters are initialized to the values present in Table 3.12. As this model is a general abstract model, the parameter values are selected randomly.

However, the model's robustness to varying parameter values is considered as important. Therefore, an analysis is performed on varying parameter values of the key parameters. Moreover, a sensitivity of the parameters "Expectation Growth Rate", "Competition" and "Need Probability" to the results of the simulation is analyzed.

Table 3.12: Initial Parameter Values Used in the Simulation

Parameter	Default Val.
customerN	10
providerN	10
K	2
C	2
noOfStates	4
noOfUseContexts	1
noOfCustomers	1000
noOfProviders	2
cusAdequateMargin	0.4
proAdequateMargin	0.4
addCompetition	FALSE
cusChngblAttPercentage	0.8
proChngblAttPercentage	0.8
Need Probability	0.25
Loyalty Biasness	0.75
Trust of Conformity	0.25
Loss of Unconformity	0.4
Innovation frequency	10
Innovation Sample Size	50
Message Transmit Percntage	0.25
Lambda (Gain)	6
Lambda (Loss)	10
Beta	50,000
Expectation Growth Rate	0.05
Trust Threshold	0.25

3.3.2. Input Data

The model does not use any external input data to represent time varying processes.

3.3.3. Sub Models

This section of the thesis contains the details of the sub models in the processes explained in 3.1.3. The descriptions of the sub models are organized under the name of the processes mentioned in 3.1.3. This section is mainly important for any future attempts to replicate the model.

Building the Model Context: Building the model context involves a sequential process as depicted by Figure 3.10. The first four sub models belong to the tasks of the Context Builder where as the last two sub models belong to the tasks of the Controller agent. The context builder is responsible for creating the structure of the artificial markets such as creating the value propositions, building up the utility landscapes and constructing the layout of the agent distribution. The controller is an agent, which acts as a mediator in the communications and interactions between the agents of the system as well as between the users and the system.

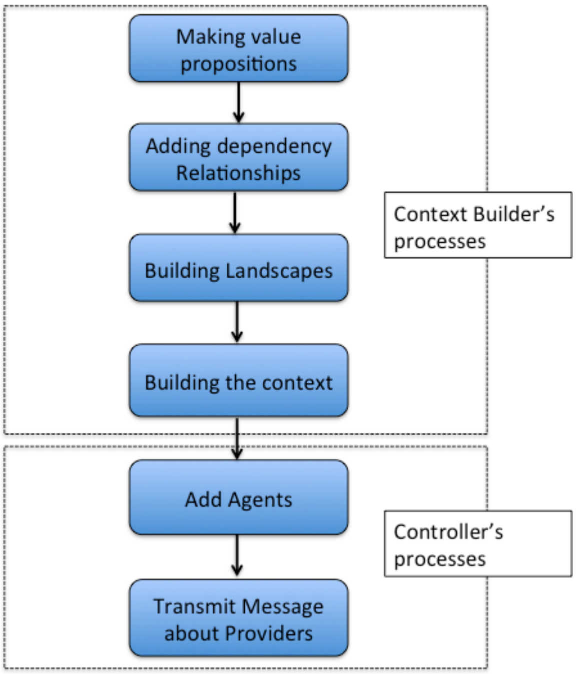

Figure 3.10: Sub Models of Building the Model Context

Making the value propositions of the two entities involves creating two lists that hold instances of the Value Item entity. Each value item corresponds to a position in the value proposition, which determines whether it is changeable or not. Furthermore, in the case of provider entity's value proposition, each value item corresponds to “basic”, “excitement” and “performance” features of the service. Thus, it is necessary to determine the type of the attribute at the time of creating the respective value item. The following procedure explains the process of making value propositions.

```
PROCEDURE (Make Value Propositions)

    Read variables customerN, providerN
    Make variables List<ValueItem> cusValueItems, proValueItems

    //Making customer’s value item list

    Determine no of changeable Positions on value proposition
    Add value items to cusValueItems

    //Making provider’s value item list

    Determine no of changeable Positions on value proposition
    Determine no of Basic, Excitement, Performance attributes


    Add value items to proValueItems

END PROCEDURE
```

The next task after making the value propositions is imposing the dependency relationships on each value item. The following procedure explains the sub model of imposing dependency relationships on each value item in the customer entity’s value proposition based on Kauffman’s NKC architecture. Adding the dependency to the value items in the provider entity’s value proposition is similar to this procedure. In this procedure, each value item in the customer entity’s value proposition is set to be depending on K other attributes of its own as well as C other attributes in provider entity’s value proposition.

```
PROCEDURE (Make the dependency relationships – Customer entity)

    Read variable List<ValueItem> cusValueItems
    Make variable List<String> customerValueItemIds,
           providerValueItemIds
    Read parameters K, C

    Fill customerValueItemIds and providerValueItemIds with IDs
```

```
    //Setting up internal complexity – K

    For each value item in cusValueItems
          Add K number of customerValueItemIds as dependency items

    //Setting up external complexity – C

    For each value item in cusValueItems
          Add C number of providerValueItemIds as dependency items

END PROCEDURE
```

Building the value propositions with value items and their relationships enables to create the utility landscapes of the two entities – customer and provider. Due to the dependency relationships imposed on each attribute of a value proposition, there can be $D^{(1+K+C)}$ number of state value combinations associated with each value item position (D = the number of states). Moreover there is a randomly drawn utility value associated with each state value combination of each value item for each context. Notably, this thesis considers only one use context for all agents. Therefore, a utility landscape contains the utility values for each state value combination for each value item position. A state value combination is represented as a string in which the last digit is reserved for the current use context. The procedure of making the landscape of the customer entity is explained below. The procedure of making the landscape of the provider entity is similar to that of the procedure of the customer entity.

```
PROCEDURE (Building the utility landscape of the customer)

    Read parameters noOfStates, K, C, noOfUseContexts
    Make variable List<String[]> customerLandscape

    Determine total no of State-Value Combinations

    For each state-value combination
          Make a state string
          For each use context value
            Add use context value to the end of state string
            Make variable String[customer + 1] landscapeItem
            Set landscapeItem[0] to state string
            For each position i on value proposition
               Determine a utility value u
               Set landscapeItem[i] to u
```

```
            Add landscapeItem to customerLandscape
END PROCEDURE
```

Finally, the context builder creates the Controller agent embedding the value propositions and utility landscapes of customer and provider entities as well as a grid layout to distribute customer agents. Finally, the context builder adds the controller agent to the context and returns the context.

When the controller is created, the controller agent adds the required number of customer agents and the provider agents to the context. Adding an agent requires making the current state of the agent as a state value combination with a length equivalent to the length of the respective value proposition. The following procedure explains the process of adding a customer agent. Adding a provider agent is also similar.

```
PROCEDURE (Adding agents to the context)

    Read parameter noOfCustomers, noOfUseContexts
    Read variable List<ValueItem> proValueItemList, cusValueItemList
    Read variable context thisContext
    Read variable List<Customer> availableCustomers
    Set count To 0
    Repeat While count < noOfCustomers
      Make Customer C
      Add C to current context
      Add C to availableCustomers
      Set count To (count + 1)
END PROCEDURE
```

After making the providers, the controller agent distributes messages about available providers to all customer agents. The customer agents can receive the messages based on the parameter *messageTransmitPercentage*, which acts as a determinant whether a given customer receives the message or not.

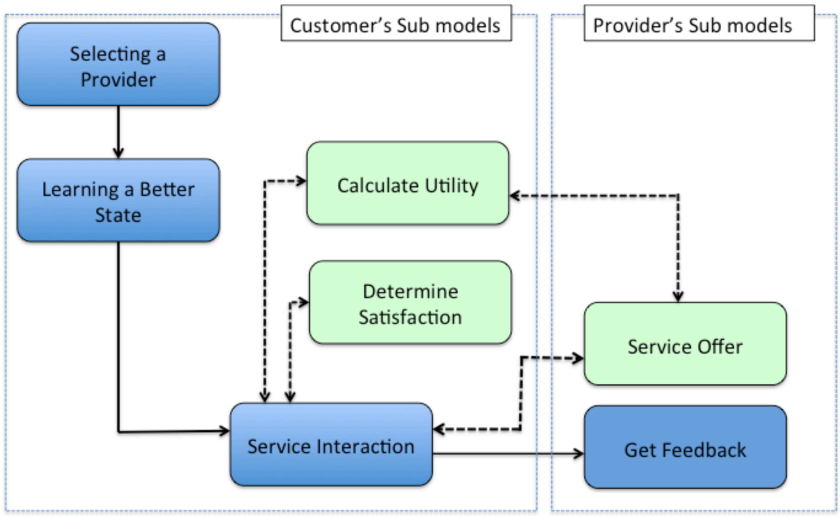

Figure 3.11: Sub Models of the Market Process

Sub Models of the Market Process: The market process involves the sub models of both customer and provider entities. Figure 3.11 depicts the sub models of the process and their flow.

The process starts with selecting a provider based on a service need. The following procedure explains the sub model. Notably, when the customer agent has no known providers available, it calls the *TellMeProviders* sub model on the neighboring customers. The neighbors are determined by the neighboring cells of the current cell of the customer agent on the grid.

```
PROCEDURE (Selecting a provider)
   Read parameters loyaltyBiasness, needProbability
   Read variable List<Customer> myNeighbors
   Read variable List<Provider>providerList
   Make variable selectedProvider
   Set selectedProvider To Null

   If needProbability > Random(double(0,1))
      If no known providers available
        Ask neighbors about available providers
        If got to know few providers
           Set selectedProvider to a random provider
      Else
        If loyaltyBiasness > Random(double(0,1))
           For each provider in the list
                Select providers with positive affection
                If no providers with positive affection
```

```
                Set selecteProvider to a random provider
             Else
                Calculate loyalty share of each provider
                Set selectedProvider probabilistically
        Else
           Set selectedProvider to a random provider
END PROCEDURE
```

Learning a better state depends on whether the customer is involved or not. An involved customer attempts to learn a better state to co-create value after identifying a provider for the service, by comparing its satisfaction in last experience with the selected provider against the satisfactions of its neighbors with the same provider. The learning involves identifying the neighbors who are more satisfied and moving to the direction of the nearest neighbor with better satisfaction by one point. In other words, the customer agent reduces the distance by moving to a one-mutant neighboring state. The sub model of learning of involved customers is explained in the below procedure. Notably, in order to calculate the distance between the current states of two customer agents, the procedure calls the *CalculateDistance* method of the Customer entity, which determines the Cartesian distance between the two state vectors.

```
PROCEDURE(Learning better states)

   Make variable myLastExperience
   Make variable myExpectedSatisfaction
   Make variable nearestCustomer

   If customer is involved with the provider
      Set myLastExperience to the last experience with selectedProvider
      Set myExpectedSatisfaction to the satisfaction of last experience

      Set nearestCustomer to the customer with nearest profile who've
      had better experience with the selectedProvider

      If nearestCustomer is not Null
            Move to a one-mutant neighboring state that reduce distance
            with nearestCustomer
End PROCEDURE
```

Service interaction is the core of all sub models. It includes several other sub models such as *calculate utility*, *determine satisfaction*, and *service offer*. The following procedure explains the sub model of service interaction. Notably, the creation of a service experience involves the update of customer's *trust* whereas a recommendation received updates the *recommendation strength*.

```
PROCEDURE(Service interaction with the selected provider)

   Make variable myServiceExperience
   Set myServiceExperience to experience with the selected provider

   If experience is a rejection of service request
        If a state change had occurred
           Return to previous state
           Record the experience as Not Accepted
           Update affection accordingly


         Else
            Record the experience as Accepted
            Calculate satisfaction
            Update delight in the experience
            Add experience to my affection with the provider

            If satisfaction is positive
                Make recommendations to neighbors about the provider
            If a state change had occurred and satisfaction is less _
                                                     than expected
               Return to the previous state

END PROCEDURE
```

A service interaction begins with a service offer made by the customer to the selected provider. Once a service offer is received, the provider uses a set number of check attributes in the customer agent's value propositions to determine whether to accept the offer or not. For example, most hotels require their customers to bear a valid credit card at the time of booking as a security. In this model, the number of such check attributes is set to a constant value of three. Based on the utility of the provider from those three attributes, the provider decides whether to accept the offer or not. The following procedure explains the sub model.

```
PROCEDURE(Service Offer)

   Make variable fitnessToServe
   Read variable myCheckAttributes
   Calculate fitnessToServe for myCheckAttributes

   If fitnessToServe > 0

      Set agreeToServe to TRUE
      Make new transaction
      If customer relationship exist already
            Add transaction
      Else
            Make new customer relationship
            Add transaction
            Add relationship to my relationships
   Else
      Set agreeToServe to False

   Return agreeToServe
END PROCEDURE
```

Calculating the utility of a value item involves calculating the utility contribution of a given value item to the overall utility of a value proposition from a given service interaction. This sub model takes the value item being considered of the value proposition of the provider entity and the current state of the provider agent with whom the customer agent has started the particular service interaction, as inputs. The details of the sub model are given by the procedure below. Based on the position of the value item in the value proposition and the dependency relationships it has, the procedure determines the relevant state value combination to get the respective utility value from the utility landscape.

```
PROCEDURE(Calculate utility of a value item)

   Read variable thisItem
   Read variable providerState
   Read variable myCurrentState
   Read variable myCurrentUseContext
   Make variable stateValueCombination
   Make variable utility

   Determine stateValueCombination
   Set stateValueCombination to stateValueCombination + _
                                          myCurrentUseContext
   Set utility from landscape
```

```
    Return utility

  END PROCEDURE
```

The utility calculated from a value item is the determinant of the satisfaction from that item. The sub model of calculation of satisfaction from utility involves determining the satisfaction based on the classification of provider's attributes as *basic, excitement* and *performance.* Figure 3.8 depicts the algorithm that explains how to determine the satisfaction based on this classification. The following procedure explains the sub model of calculating the satisfaction from utility.

```
  PROCEDURE(Calculating satisfaction from utility)

    Read variable thisItem
    Read variable utility
    Read variable currentExpectation
    Make variable difference
    Make variable satisfaction
    Make variable category

    Set difference To (utility – currentExpectation)
    Set category To thisItem's category

    If category = "Basic" And satisfaction < 0
      satisfaction = difference
    Else
      If category = "Excitement" And satisfaction >= 0
       satisfaction = difference
      Else If category = "Performance"
       satisfaction = difference

   Return satisfaction

  END PROCEDURE
```

Customer agent's feedback is the end of an iteration of the market process. Once received a feedback, the provider agent evaluates its utility with the service interaction and updates its memory on the relationship with the particular customer agent. The following procedure explains the sub model of receiving a customer feedback.

```
PROCEDURE(Get Feedback)

   Read variable customerID
   Read variable customerDelight
   Make variable thisRelationship
   Make variable thisTransaction


   Find thisRelationship with customerID
   Set thisTransaction to last transaction of relationship
   Set delight of thisTransaction to customerDelight
   Calcuate my utility of the transaction
   Set myUtility of thisTransaction to calculated utility

END PROCEDURE
```

Apart from the sub models related to the main processes explained in this section, there are other methods of each entity that support the functionality of the model. Appendix D contains the extended pseudo codes of the sub models explained in this section.

4. Simulation Experiments and Results

This section contains the details of the basic experiments conducted using the agent-based model explained in Chapter 3.

4.1. Dynamic Behavior of Customer Loyalty

The first experiment involves the investigation of the dynamic behavior of customer loyalty under default conditions mentioned in Table 3.12. These default conditions match to an untapped market with limited number of providers and lower customer expectation levels. Furthermore, since the *need probability* is set to 0.25, this could be considered as a market with less regular needs. The corresponding graph, which displays the dynamics of loyalty over a period of one thousand time steps, is available in Figure 4.1.

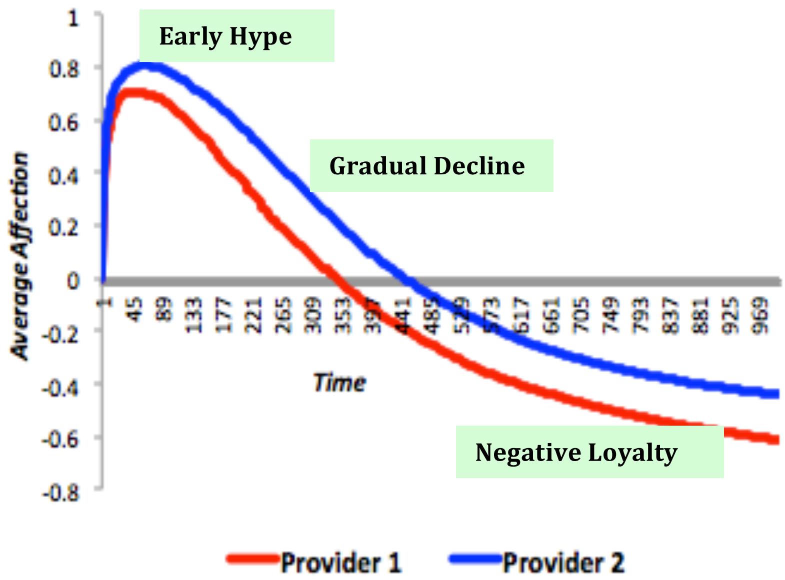

Figure 4.1: Dynamic Behavior of Customer Loyalty with Time

Here, the red and blue lines correspond to the average affection entertained by the two service provider agents in the system, the Provider 1 (P1) and Provider 2 (P2) respectively. The average affection of a given provider at a given time involves the average of the affective commitments of all customers who have used the service of that provider so far. Since the affective commitment is explained in section 3.2.1 as the determinant of customer loyalty, the variation of affective commitment over time reasonably depicts the dynamics of customer loyalty.

The graph in Figure 4.1 has three distinguished phases; an early hype, a gradually declining phase and a negative loyalty phase. The early hype reflects a higher level of customer satisfaction at the early stages of the market with lesser expectations. However, as the expectations grow with initial delightful customer experiences, loyalty declines gradually as the experiences would not be as delightful as before with elevated expectation levels. Once the expectations are grown beyond the levels that the service providers could achieve, loyalty moves to the negative phase.

4.1.1. Dynamics of Loyalty and Customers' Switching Behavior

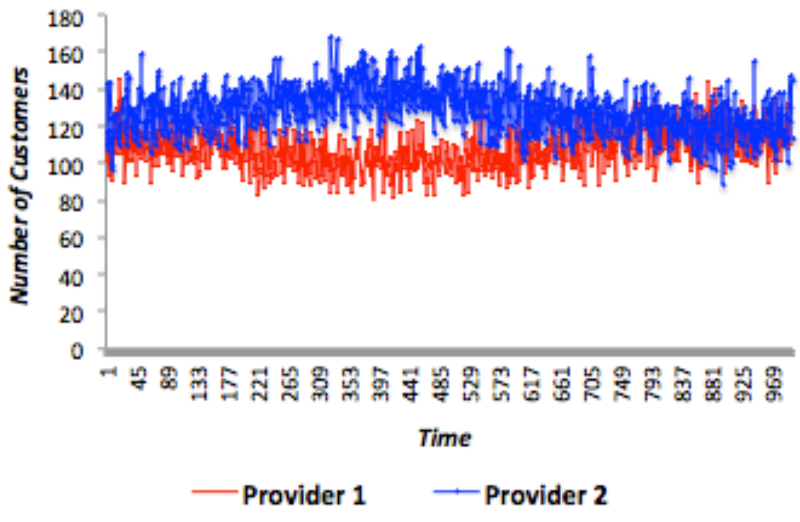

Figure 4.2: Number of Customers Arrived for Each Provider over Time

According to the current decision rules of customer agents, customers do not give up a service need in case if a provider, to whom they are loyal, couldn't be found. In other words, when a provider could not be selected based on loyalty, customers make a random decision to pick up a provider out of all providers known to them. Therefore, despite the dynamics of loyalty explained previously, the market shares of each provider remains stable by the end of the simulation as depicted by the chart in Figure 4.2.

However, analysis of the frequency of individual switching throughout the simulation run, which is presented by Figure 4.3, reveals a different dimension of this consistent market share. Figure 4.3 reveals an exponential growth pattern in the frequency of switching of customer agents between service providers. This implies a relationship between customer loyalty and their switching behavior. In other words, when the customers' loyalty towards providers is lower, they keep on switching between available providers more frequently.

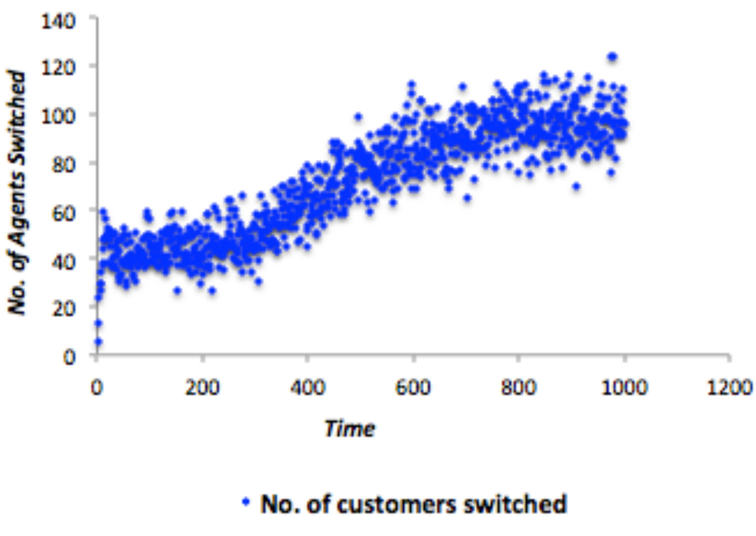

Figure 4.3: Frequency of Customer Switching over Time

This exponential growth of the frequency of customers' switching over time implies a reasonable pattern in the real life. As a service reaches its maturity, customers' expectations become elevated up to a level, at which making them delighted is challenging. Hence, none of the service providers become "special" to the customers leading customers to make purchase decisions randomly than based on their loyalty. In other words, level of customer expectations has a significant role in the dynamics of customer loyalty. The following graph in Figure 4.4 depicts the change of the frequency of customers' switching when the initial customer expectations were set as high as 70%. According to that, at higher initial expectation levels, customers display higher frequencies of switching from the beginning.

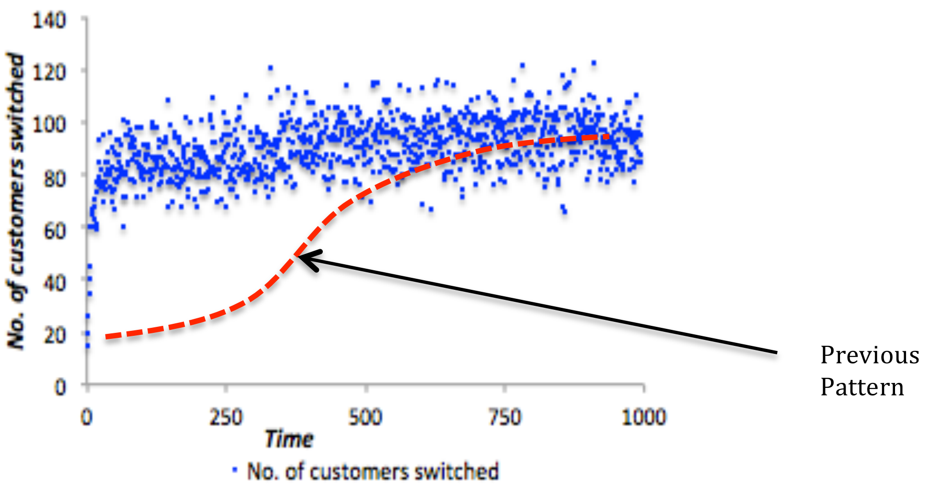

Figure 4.4: Frequency of Customer Switching Over Time at Higher Initial Expectations

4.1.2. Robustness of the Result

The model gives similar results to changing parameter values of *N, K, C, No of States, No of Customers* and *No of Providers*. Furthermore, it gives similar results for different random seeds. However, with higher values assigned to the above parameters, the computational complexity increases and the system demands for higher computing resources. For example, testing the system for more than 2000 customers was obviously stressful to a single personal computer. Moreover, the output is not complying with this standard pattern when the initial expectations of customer agents are set at a higher level such as 0.8. This is quite predictable as the room for customer satisfaction and delight is limited when the expectations are at higher levels.

4.2. Sensitivity to Competition

Typically, loyalty is expected to keep customers stuck to a particular provider in the long run. Therefore, one would expect customers to be immune to new arrivals of providers to the market, at least during the early stages of the simulation where the loyalty is at hype. The simulation provides interesting, yet contradictory, insights about this phenomenon. This section contains the simulation results of customers' response to competition when a new competitor is added during the simulation run.

4.2.1. Early Vs. Late Competition

It is interesting to see the customer agents' move when there became a new competitive provider available when their loyalty towards the existing providers is at growth. Figure 4.4 – (a) depicts the change of average affection with a new competing provider agent being added at time = 50 whereas Figure 4.4 – (b) depicts the corresponding change in market share.

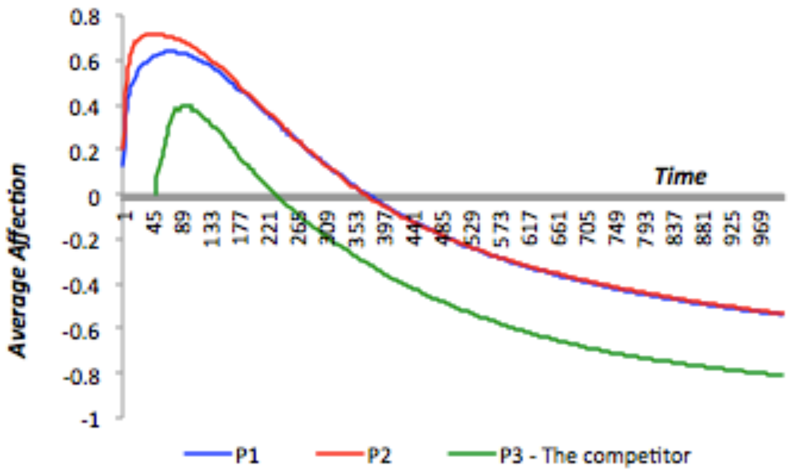

Figure 4.5 – (a): Change of Average Affection with a Competitor Added at Time = 50

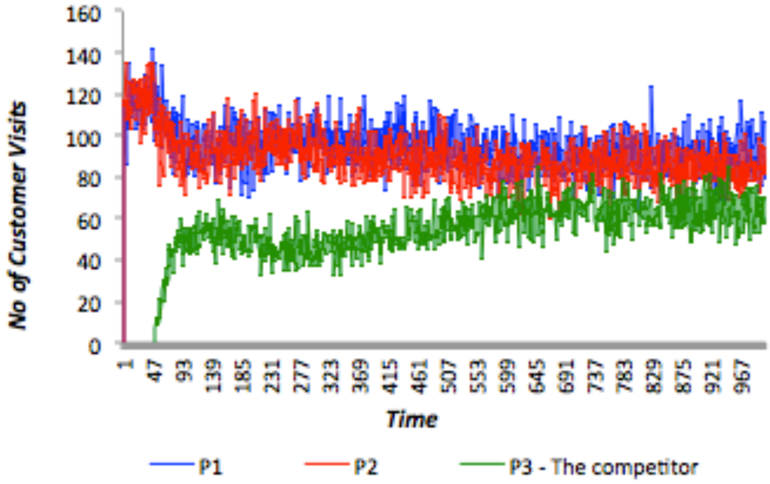

Figure 4.5 – (b): Change of Market Share with Early Competition

Even though it could have been predicted that the customers would not switch when their loyalty is at hype, Figure 4.4 – (a) indicates that it is still possible for an early competitor to secure a significant customer loyalty. Furthermore, Figure 4.4 – (b) shows that the new competitor is able to eat into a significant portion of the market share the initial service providers had been enjoying. This counter intuitive result could be analyzed with respect to the customer expectations. Since the initial customer expectations of the customer agents were set at a lower level (< 0.4), there is ample room for the providers to delight their customers. Therefore, a competitor who arrives earlier to the market still get a chance to offer a delightful service to the customers and secure a significant market share.

One would have doubts on this results as a decision of a competitor to enter a new market, in which existing providers perform well, would not be taken randomly but after a careful assessment of the value propositions of existing competitors. For example, a new entrant would consider either imitating the best performing rival or coming up with an innovative value proposition to compete with the existing rivals. Graph in Figure 4.6 depicts the change of average loyalty when a new competitor enters a market early, imitating the best performing service provider. According to the graph, there is no significant change in result compared to Figure 4.5 – (a).

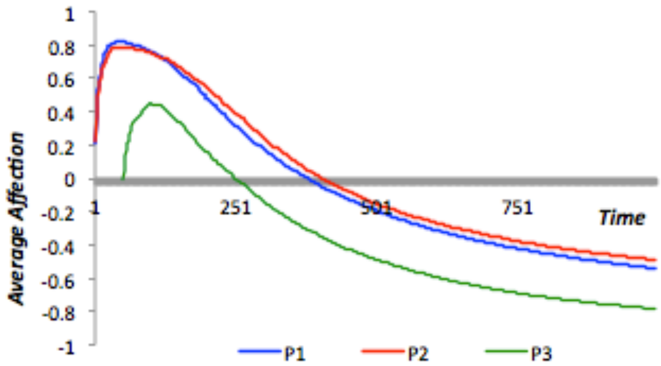

Figure 4.6: Change of Average Affection with a Competitor Imitating the Best Existing Provider Added at Time = 50

This result is further corroborated with the analysis of the performance of a competitor who arrives when the average loyalty of existing providers is declining. Figure 4.5 – (a) depicts the performance of a provider agent added at time = 200, when the average affections of the two existing providers were declining. This suggests that the new provider has not got enough room to delight customers to secure a growth in customer loyalty.

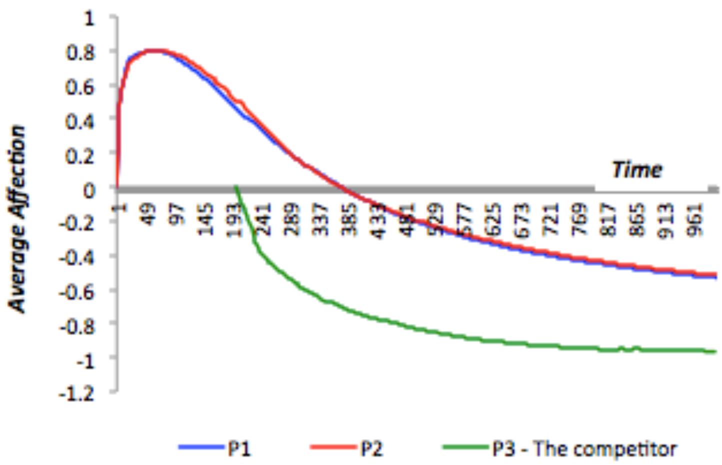

Figure 4.7 – (a): Change of Average Affection with a Competitor Added at Time = 200

However, Figure 4.7 – (b) shows that the new competitor has still been able to secure a market share, even though comparatively smaller, despite a decline in average affection. Investigation into the individual decisions of customer agents who either switched to P3 or retained with P3 proves that the initial growth in market share is due to the customers who still had a positive affection with P3. In other words, those individuals who still have a positive affection with P3 contribute to the spread of word of mouth regarding the newly arrived provider. However, market share in the later part of the simulation is mainly due to the random moves of customers as none of the providers are able to delight customers due to elevated expectations among customers.

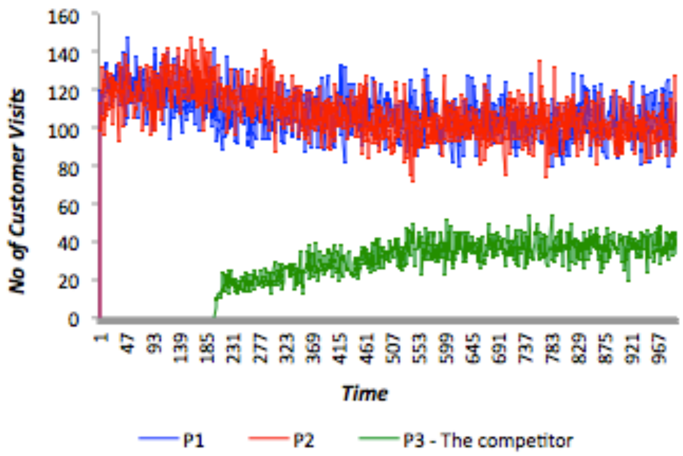

Figure 4.7 – (b): Change of Market Share with Late Competition

4.2.2. Impact of Expectation Growth

The parameter *Expectation Growth Rate* remained constant for all simulations in the previous subsections at 0.05. This value corresponds to a growth rate of 5% and was selected arbitrarily. The previous sub section suggested that the lower customer expectation levels might be enabling competitive providers, who arrive early into the market, to gain customer loyalty and market share despite the customers' loyalty towards the already existing providers being at hype. In support to that argument, this section investigates the impact of the expectation growth rate on the competitive power of newly arriving providers. Figure 4.6 – (a) depicts the change of average affection with a new provider added at time = 50 and with expectation growth rate set to 0.01. Moreover, Figure 4.6 – (b) depicts the corresponding change in the market share.

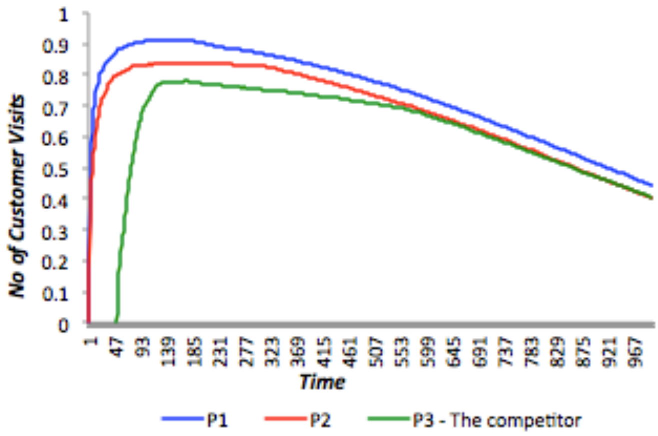

Figure 4.8 – (a): Change of Average Affection with Low Expectation Growth Rate (=1%) and a Competitor Added Early (time = 50)

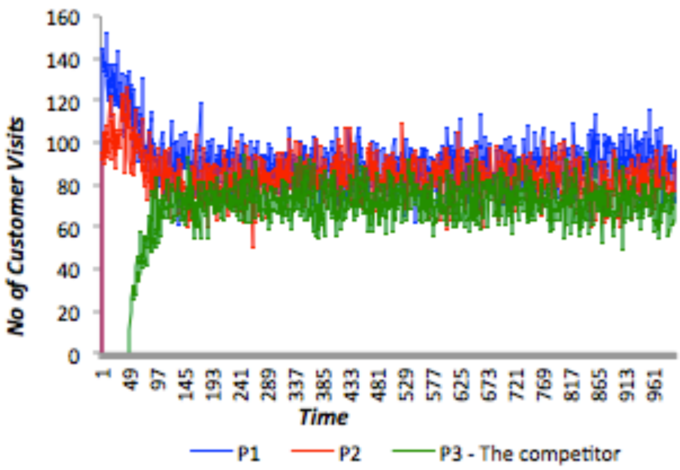

Figure 4.8 – (b): Change of Market Share with Low Expectation Growth Rate (=1%) and a Competitor Added Early (time = 50)

Notably, the average affection of all providers, as depicted by Figure 4.6 – (a), continues to be positive even after 1000 time steps. Moreover, the competing provider – P3 – enjoys almost same average affection as the two older providers P1 and P2, while maintaining almost equal market share as well (figure 4.6 – (b)). This strengthens the idea that the expectations levels of the customers play a crucial role in customers' switching.

4.2.3. Impact of Loyalty Biasness

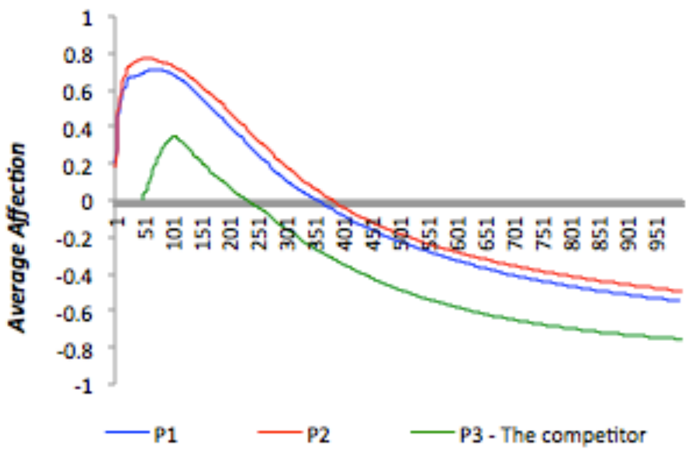

Figure 4.9 – (a): Change of Average Affection with Higher Loyalty Biasness (=90%) and a Competitor Added Early (time = 50)

The default parameter value for *loyalty biasness allows* a 25% chance for a customer agent to select a service provider randomly bypassing loyalty. However, the result of the simulation does not change significantly even if the biasness to loyalty increased to 90%, giving only 10% chance to select a provider randomly. As depicted by Figure 4.7 – (a), even when the loyalty biasness is set to 90%, an early competitor can receive significant average affection. Moreover, Figure 4.7 – (b) shows that the new comer can secure a significant market share despite customers are more biased towards loyalty.

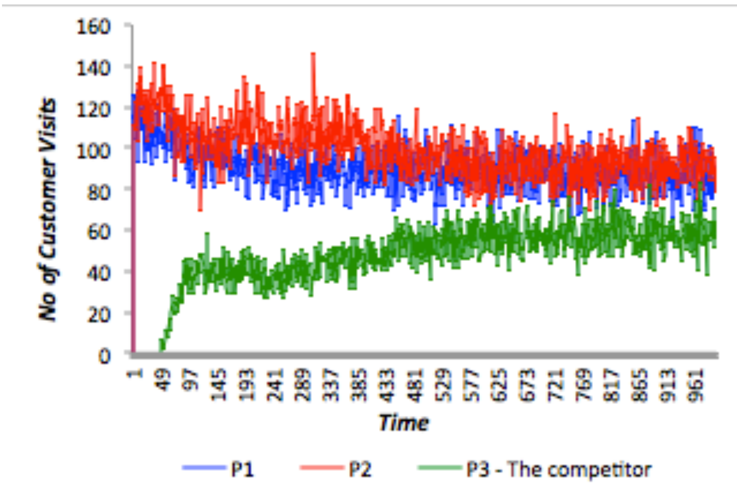

Figure 4.9 – (b): Change of Market Share with Higher Loyalty Biasness (=90%) and a Competitor Added Early (time = 50)

5. Discussion

This section of the thesis contains a general discussion about the objective and significance of the model. Furthermore, it contains a discussion on a potential application of the model.

5.1. About the Model

5.1.1. What this Model Is?

This model is a typical abstract agent-based model and does not represent any particular empirical situation. However, it could be easily customized and applied to real world applications. In fact, according to Axelrod, ABM does not necessarily aimed to provide an accurate representation of a particular empirical application. Instead, the goal of ABM is to enrich our understanding of fundamental processes that may appear in a variety of applications [72].

Being an abstract model, it serves for a set of important purposes other than prediction. According to Epstein [73], there are about sixteen purposes of modeling other than prediction namely 1.) Explain 2.) Guide data collection 3.) Illuminate core-dynamics 4.) Suggest dynamical analogies 5.) Discover new questions 6.) Promote a scientific habit of mind 7.) Bound outcomes to plausible ranges 8.) Illuminate core uncertainties 9.) Offer crisis options in near real time 10.) Demonstrate tradeoffs / suggest efficiencies 11.) Challenge the robustness of prevailing theory through perturbations 12.) Expose prevailing wisdom as incompatible with available data 13.) Train practitioners 14.) Discipline the policy dialog 15.) Educate the general public 16.) Reveal the apparently simple (complex) to be complex (simple).

5.1.2. Uses of the Model

Out of the numerous purposes of modeling, uses of this model could be discussed in relation with five major purposes.

Explain: This model can be considered as a starting point for a range of consumer research that focus on loyalty and switching behavior from the service-dominant logic perspective. As the mindset of goods-dominant logic is prevailing yet, it is necessary to utilize different techniques to help understanding the concepts of service-dominant logic and transform them into a set of operational features of a market. In that sense, this abstract model helps understanding the co-vocabulary of service-dominant logic and service science, and explaining the functionality of a market through the lens of service-dominant logic. For example, the agent-based approach helps representing a market as a system of interacting entities based on the essence of viable systems approach and service system abstraction. Furthermore, it formalizes customer loyalty as stemming from customers' emotional responses to consumption situations by emphasizing the link between the concept of value co-creation or value-in-use in service interactions and customers' emotional responses to consumption situations that lead to loyalty. In other words, the model focuses on interactions between system entities rather than the entities it self to explain loyalty as a dynamically changing property.

Illuminate Core Dynamics: Even though the model is abstract, it successfully illuminates the core dynamics of loyalty in a competitive market. The market simulated by this model corresponds to a fresh and untapped market, in which customer expectations are very low. In such markets, an arrival of service provider(s) is more likely to create fuzz among customers, letting customers to be delighted, making them affectively committed and involved with the provider(s) as well as making them actively engaged in positive word-of-mouth. However, due to increasing customer expectations, delight is not eternal. This affects the level of value co-creation in service interactions and thereby the perceptions of customers about providers and their referrals. Moreover, the affective commitment of customers towards providers gets perturbed with arrivals of competitors to the market as they get more choices. Therefore, the affective commitment of customers towards providers, which determines the loyalty according to Bowden [3], varies over time. This model successfully captures these core dynamics, which are explained in Chapter 4.

Illuminate Core Uncertainties: The model is also capable to illuminate the core uncertainties. That is, the dynamics of loyalty is sensitive to few parameters of the system. For example, there is always a chance for ta given customer to make a random decision to select a provider irrespective of his or her loyalty towards that provider. On the other hand, there are uncertainties involved with time to market. For example, when a market is matured with elevated and established customer expectations, a new coming provider may find it difficult to position itself in customers and secure a significant market share without substantial innovation. Being abstract, the model still supports investigating such uncertainties indicating its appropriateness to extended studies alike.

Provoke Questions: Another advantage of the model is that it enables asking questions about the research work being done. As this model could be claimed as the first of its kind, generating new questions is also a use of the model. For example, according to goods-dominant logic, money is exchanged for a good (or its intangible counterpart, service) in a transaction between a provider and a customer. However, money is an operand resource according to service-dominant logic and its role in customer-provider interactions is not very prominent, even though its involvement in the interaction is apparent. On the other hand, there is a difference between customer retention due to loyalty and customer retention due to lock-in conditions [74]. Moreover, there could be other restrictive conditions involved in a market such as costs incurred in switching decisions as well as barriers present in some markets for new providers to enter or imitate [75]. These types of questions may not be apparent at the first phase of the model construction. On the other hand, they may not be important unless the model is being applied to a specific industry. However, the fact that this model gives rise to such questions is an advantage as it helps the progress of research following the path of this research.

Support Data Collection: Finally, this type of abstract model would guide data collection. For example, the results of the model show that even when the loyalty level of providers are at peak, an early competitor, who enters the market when customer expectations are not grown higher, can secure significant level of loyalty as well as market share. This is an interesting and counter intuitive result generated from the model, which may worth further investigating into. In this endeavor, the model plays the role of a guide to determine which data is to be collected from real world processes in order to support this investigation. On the other hand, when this model is applied to a specific domain, it requires further data collection. For example, when applied to a specific domain, all attributes of a value proposition, their dependencies, states and respective utility values become specific to that particular domain. This entails the necessity of valid parameter estimation with respect to the particular domain. In that endeavor too, this model may be useful to decide which data needs to be collected and how it should be collected.

With these uses, the proposed model and its approach could also be effectively used as a tool for service designers following the soft systems approach [76]. There, it is possible to develop value propositions, interrelationships and corresponding utilities in conversation with potential stakeholders with the ultimate objective of enabling successful value co-creation for the customers.

5.1.3. What this Model Is Not?

The model presented in this thesis need not to be confused with a facsimile model [77] aiming at prediction. The main purpose of this model is to enrich the understanding of the fundamental process of customer loyalty by initiating a discussion on its dynamic nature from the service-dominant logic perspective. Therefore, more emphasis is given to formalize an abstract computational model that can be used as a concrete basis for future facsimile models applied to different industries. Therefore, prediction is the ultimate target of initiating this research path, but not an objective of the research presented in this thesis.

5.2. Potential Application – Tourism Sector using Sri Lanka as the Test Bed

5.2.1. Why Tourism Domain?

Intention to revisit a destination[1] has been viewed as an important research topic in both academia and the tourism industry and it has been recognized the importance of observing tourists' revisit intention from a time perspective because the intention often changes over time [78]. Research on destination loyalty shows that one of the most decisive factors in a further visit to a destination by tourists is their satisfaction with previous stays there [79][80]. It is well established in tourism literature that both overall tourist satisfaction and a tourist's intention to return are particularly determined by his/her assessment of the destination's different attributes [37]. In other words, the evaluation of individual attributes of a destination affects customer satisfaction. Therefore, a study on the dynamics of loyalty in tourism is significant and an approach based on the satisfaction coming from evaluations of individual attributes seems to be acceptable.

On the other hand, service-dominant logic is changing the attitudes and approaches of the tourism service managers and the academics. According to Shaw et al., service-dominant logic is particularly relevant to tourism management since it 'is based on an understanding of the interwoven fabric of individuals and organizations' [81]. Tourism is highly driven by expectations and experiences [82][83], and as such suppliers and consumers interact more closely together at all stages of their relationship [81]. In other words, tourists co-create experiences in interactive processes [84]. In this context, the application of the concepts of service-dominant logic provides a framework with which to examine supplier-customer processes involved in co-creating the visitor experience [81]. Moreover, the quality of the experience offered by a tourist destination is more than the sum

[1] Meaning of the word destination depends on the level of aggregation. As far as this thesis is concerned, it includes service provider organizations such as resort hotels too.

of its parts; it depends in important ways on how the organizational parts are interconnected, the way they act and interact and the relations between the actors involved [85]. This highlights the usefulness of the systems approach as well as the applicability of NKCS architecture based models to the study of dynamics in tourism markets.

Apart from the customer satisfaction based measures of customer loyalty, customer referrals too matter a lot in the tourism sector in bringing up (or down) the brand images of tourism service providers. For example, traveller oriented web sites such as TripAdvisor [86] offers platforms for customers exchange their recommendations and opinions about tourism service providers/destinations all over the world, putting the quality of the providers/destinations under a continuous review process. On the other hand, communities like travel bloggers [87] as well as free and independent travellers [88] too are actively involved in spreading their experiences at various destinations around the world through web medias. These active and fast information exchanges are more likely to influence customers' perceptions about service providers in tourism markets. Therefore, destination loyalty in tourism markets needs to be evaluated dynamically through a study of the interactions between system components.

5.2.2. Sri Lanka as a Tourism Market

Sri Lanka has a long history as a tourism market. Due to its utmost scenic beauty, cultural heritage from over 2500 years, sandy beaches, rich wildlife and stronger historical relationships with Asian and European countries, Sri Lanka was a well-known and popular tourist destination among tourists all over the world. However, the country's tourism industry was badly hit by the political issues between the government of Sri Lanka and the north-based rebels, which led to numerous terror attacks and several wars. The military defeat of the terrorists in May 2009 was a relief to the tourism industry, which regained hopes of a rapid growth. In fact, making Sri Lanka a regional tourism hub was a key element of the country's development strategy [89].

The end of war opened up not only hopes but also a vast land and coastal area, which was not open for tourism in the past couple of decades. However, the country is still lacking the necessary infrastructure for a booming tourism sector. Therefore, Sri Lanka's tourism market after 2009 could be considered as a new and growing market. Figure 5.1 shows the growth of the market in 2014 compared to 2013 in terms of tourist arrivals in each month.

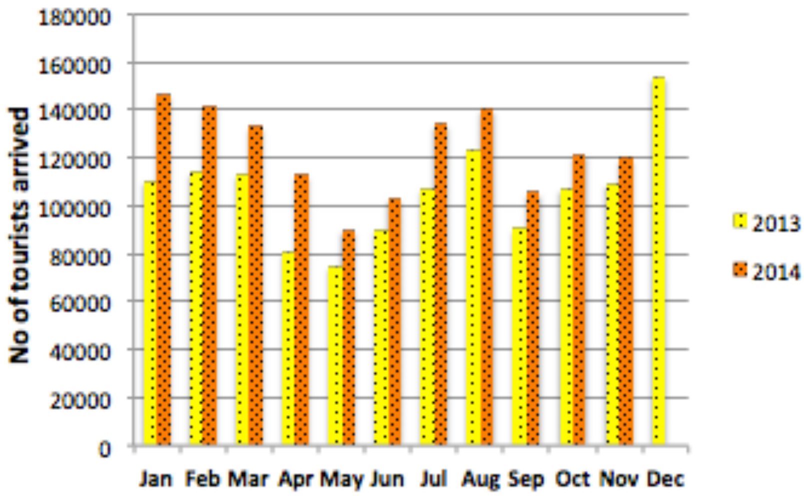

Figure 5.1: Comparison of Tourist Arrivals to Sri Lanka in 2013 and 2014

Source: [88]

In order to cater this growing market, the tourism development strategy of Sri Lanka for the period of 2011-2016 expects to attract USD 3000Mn foreign direct investment to the country [90]. Investments would largely be required to increase the number of hotel rooms available in the country as it is predicted that the country would require 45000 hotel rooms by 2016, which is almost double of the 22735 rooms existed in 2010. However, tourist destinations go through a life cycle pattern according to Butler, which is shown in Figure 5.2 [91] [92]. Therefore, for this booming industry to be sustainable and the massive target investments to be fruitful, it is necessary to avoid possible declines in the future.

One critical factor of avoiding possible declines is ensuring customers' revisits. For example, according to the airport survey conducted in 2009 on the departing tourists, a large majority of 61% was on their first visit to Sri Lanka [93]. Therefore, it is highly necessary to make attempts to transform these first time visitors to repeat visitors. On the other hand, Sri Lanka is currently experiencing growing number of tourists from non-traditional regions such as India and China, which comprise millions of potential tourists [94]. This changing diversity of customer profiles requires better understanding on customer-provider interactions in terms of co-creating positive experiences and the resulting dynamics of loyalty, for the investments to be paying back through systematic service innovation. Hence, a service-dominant logic based study would be beneficial for the investors as well as the tourism research community and government authorities. Furthermore, the model being presented in this thesis could be an interesting starting point for such a study using computational methods [95].

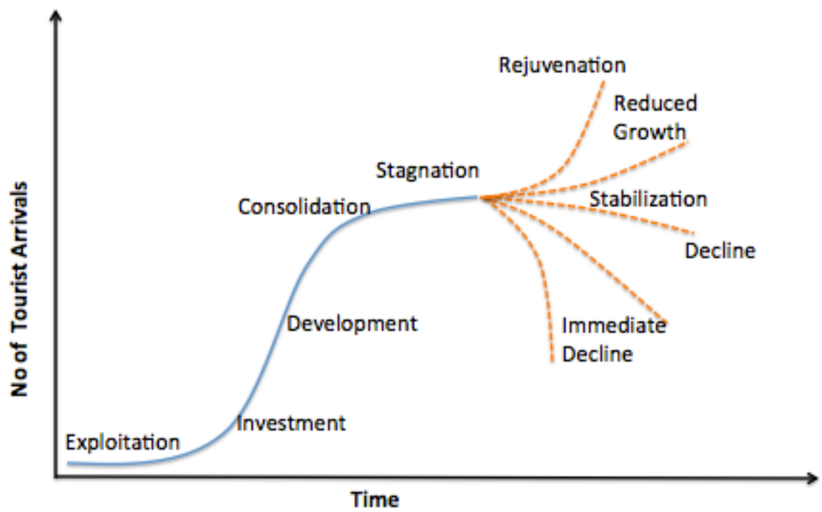

Figure 5.2: Tourism Area Life Cycle of Butler, 1980

5.2.3. Application Guidelines to a Tourism Market

A tourism market comprises numerous hotels, which could be considered as service providers in a market system. In fact, service providers in a given market could be defined at different levels of aggregation. However, since hotels does a significant impact to the customers' experience and hence the intention to revisit, this discussion considers the interaction between hotels and their customers.

A hotel is a place where a tourist can stay for one or more days, engaging in different activities in the vicinity. These activities are usually associated with the famous attractions situated around the hotel, cultural events, famous cuisines of the country or area as well as the geography and wildlife of the area. Therefore, a hotel's value proposition would include provisions to such activities. For example, a hotel in the North-Central Province of Sri Lanka would emphasis its location near the famous Sigiriya Rock, the ancient cities of Anuradhapura and Polonnaruwa, safari parks and historical tanks, and offer a range of activities to enjoy those attractions such as air balloon trips, bird watching, elephant safari, trekking and boat rides. Resources allocated for a particular provision on the value proposition reflect a particular state. For example, residents of some hotels are able to see wildlife only through safari rides where as some hotels offer air balloon trips in addition for the guests to have an aerial view, which is clearly a different and advanced state. Figure 5.3 contains a part of a hotel's value proposition and potential states.

Value Attribute	States
…	
Location	Coastal, Mountainous, Historical
Access	Walk only, Vehicle, Walk or Vehicle
Exploration Facilities	Not Provided, Ground-based, Ground plus Aerial
Attractions	Historical, Natural, Both
Internet	No Internet, WiFi Zones, Free In-room Access
Cusine	Local, Local and Traditional, Traditional and International
Spa	Massage, Ayurvedic, Specialized
…	

Figure 5.3: Part of a Hotel's Value Proposition and Potential States

Notably, these value attributes and the states could be further decomposed depending on the situation. States of all attributes of a given hotel determines the hotel's current profile.

Similarly, a tourist visiting a particular hotel too has a value proposition that comprises multiple attributes. For example, tourists' profiles may differ based on their demographics, psychographics, knowledge and skills, country of origin etc. For example, some tourists such as backpackers do not possess much cash in hand but have adventurous personalities to try out novel experiences where as some, such as mass tourists, possess more cash and look for relaxation and comfort. Thus, whether a tourist is adventurous or not is an attribute of tourists' value proposition, which could be at two states; Yes or No. Figure 5.4 contains a part of a tourists' value proposition and potential states.

Value Attribute	States
…	
Adventurous	Yes, No
Interests	Relaxation, Outdoor activities
Use Internet	Yes, No
Possess PC/Mob. Device	Yes, No
Age	<25, 26~50, > 50
Possess Credit Card	Yes, No
Cash	< 500$, 500$> & < 2000$, > 2000$
…	

Figure 5.4: Part of a Tourist's Value Proposition and Potential States

Value Co-creation: This representation helps understanding how tourists co-create value (or experience) in interactions with hotels. According to the model being discussed, a service interaction has a value for both parties. In other words, tourists with different profiles co-create different values with the same hotel whereas the hotel too co-create different values with those tourists depending on their profiles. For example, a backpacker and a mass tourist will not co-create the same value with a star hotel as the profiles of the two types have different states in attributes. On the other hand, a star hotel would not co-create the same value with a backpacker and a mass tourist. Figure 5.5 helps further explaining the process of value co-creation in a hotel-tourist interaction.

As shown in Figure 5.5, attributes on a value proposition depends on a certain number of other attributes of its own as well as a certain number of attributes of the other entity. For example, location of the hotel would be related to the means of accessing the hotel, the existence of tourist attractions in the vicinity and the exploration facilities provided. Furthermore, the location is related to the attributes such as age of the tourist, interests of the tourist and whether the tourist is adventurous or not. An equivalent computational representation using NKC architecture would set parameters K = 3 and C = 3. According to this relationship structure, value contribution of a given attribute depends not only on its own state but also the states of the other related attributes. For example, a hotel could be situated on a mountainous location with lots of attractions in the vicinity and various activities to explore the area but with access only by walk. In such a case, a young adventurous tourist with a passion to outdoor activities such as hiking and climbing would co-create maximum value with the attribute "location". However, an older tourist may find it difficult to co-create value with the location attribute of that hotel as the tourist does not possess the required resources (Ex. Health conditions).

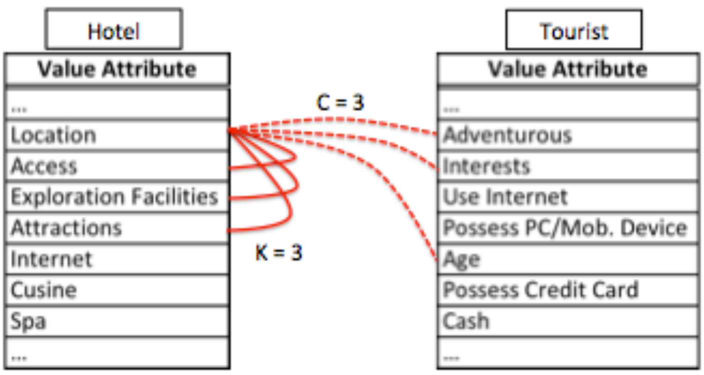

Figure 5.5: How Value Co-creation Works in Tourism

Application overview: The model presented in this thesis corresponds to a situation depicted by Figure 5.6. There are multiple hotels (resorts) at a given tourism market from which customers can select. The hotels have different profiles depending on their characteristics. There are also multiplicities of current and future customers. These hotels and customers correspond to the service system agents of the model and their current profiles correspond to the current states of the agents. Customers do service interactions with selected providers, co-create value and make recommendations to other potential tourists known to them about their experience at the respective hotel, if it is positive. For example, the customer X in Figure 5.6, who engage in a service interaction with resort B recommends resort B to customer Y, who is already a customer of resort A. This recommendation implants a positive affection in customer Y towards resort B, giving him/her another option to consider in the next visit.

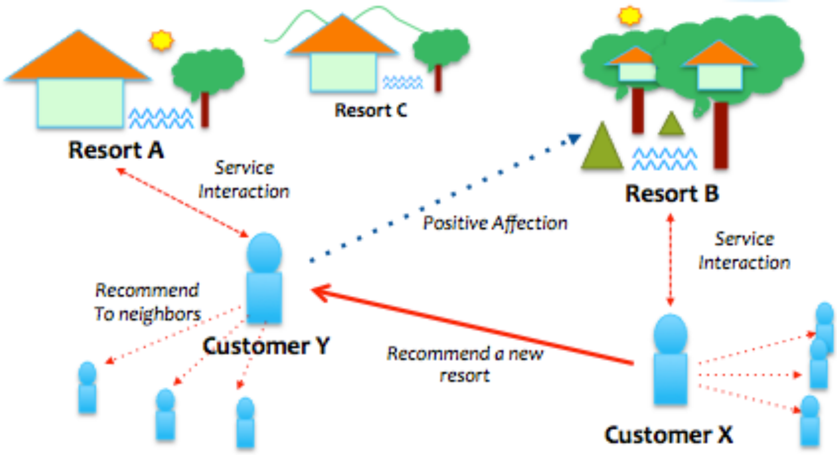

Figure 5.6: Service System Interactions in a Tourism Market

The service interaction process: A service interaction in tourism market starts with getting a service need, i.e. a place to stay during the trip. The potential tourist then seeks for a suitable hotel among the known places or the places recommended by others. Once a hotel is selected, the tourist approaches the hotel for a booking, which is a proposal for service. There the hotel would use certain check attributes to decide whether to accept the proposal or not based on the potential value of the tourist. For example, the hotel would ask for a valid credit card as a security for the booking. If the hotel accepted the proposal, the tourist would arrive at the hotel on the decided date to start co-creating value. If the co-created value were positive, the tourist would recommend the hotel to other potential tourists known. Even though this process may have slight variances, it generally corresponds to the process depicted by the figure 2.2.

Challenges: While the application of the model to the context of tourism looks interesting and straightforward, there exist some practical challenges. One major challenge is the estimation of parameter values. For example, in the current simulation, the parameters have not been estimated using any data representing the real environment. However, application of the model to the domain of tourism requires valid parameter estimation in order to get significant insights. There, parameters like “expectation growth rate” would be crucial for accurate results and the estimation of such parameters would be challenging; hence would need special attention. On the other hand, application of the model to the specific domain of tourism gives each attribute of a value proposition a meaning. Therefore, determining the relationships between attributes denoted by parameters K and C in this abstract model would have to be specifically determined by the researcher. Moreover, the utility values associated with different state value combinations needs to be realistically evaluated involving real users. Therefore, this application involves substantial fieldwork in terms of data collection. Alternatively, a “virtual grounding method” [77] could also be used to estimate parameters.

6. Conclusion

6.1. Remarks on Research Questions

This thesis reports a research attempt to study the dynamics of customer loyalty from the service-dominant logic perspective using agent-based simulation in order to answer the traditional question of why customers defect (switch). The motivation for the thesis came from the observation of increased interactions among market actors and rapid expectation growth in modern markets, which are more likely to transform customer loyalty into a dynamic property emerging from market interactions. In the study of customer loyalty as a dynamic property, this thesis mainly focuses on three research questions:

i. How does customer loyalty dynamically change in competitive business environments?
ii. How such dynamics could be effectively studied?
iii. If such dynamics exist, how are customers' switching decisions associated with them?

The thesis proposes to combine the agent-based modeling methodology, which is well known in the study of complex adaptive systems with the emerging market perspective of service-dominant logic to study customer loyalty as a dynamic property. Service-dominant logic, which is proposed as an alternative mindset to the traditional transaction-oriented mindset, provides the foundation for the study through conceptualization of market entities and interactions.

The proposed agent-based model in this thesis is a typical abstract market model. It has only two types of agents; service providers offering a single service and their customers. However, this abstractness effectively lays the foundation for a new research direction on customer loyalty based on computational modeling and service-dominant logic. It helps to explain market entities through the lens of service-dominant logic and mimic the processes of value co-creation, which are central to the service-dominant logic. Furthermore, it helps to understand how customers' emotional responses to value co-creation experiences lead to fluctuations in affective commitment making loyalty to be dynamic. In other words, the proposed model brings a range of conceptual discussions into an experimental setup on a computer, enabling further computational experimentation in the future.

The simulation experiments reveal that the customer loyalty for a given service provider follows a particular dynamic pattern. According to that, any service provider who had been in the market from the beginning would go through a pattern with an early hype followed by a gradual decline due to growing expectations. The results also reveal that a new competitor can still secure a significant market share if entered to the market early despite the existing providers going through their hypes of loyalty, or if the expectation growth rate of the market is lower.

6.2. Implications of the Research

The research presented in this thesis has implications for both researchers and practitioners. As far as the practitioners are concerned, it emphasizes the importance of managing the expectations of their customers. On one hand the lower expectation levels makes them more vulnerable to competition from new entrants and on the other hand, higher expectations make customers to make more random choices than loyalty-based choices. Furthermore, the proposed model could be suggested as an effective tool for service designers following the soft systems approach to design robust services aiming at successful value co-creation. As for the researchers, this research provides a useful starting point for service-dominant logic based research, not only on customer loyalty but also on other aspects of consumer behavior.

6.3. Limitations

The agent-based model developed in this research inevitably has its own limitations, which would be addressed in future research. Firstly, it mainly focuses on customer-customer and customer-provider relationships and does not take into account the provider-provider relationships. However, in a typical market, providers may use competitive, corporative or mixed strategies to face the competition and survive. Secondly, loyalty is clearly differentiated from lock-in situations in the marketing literature. However, switching barriers do exist in many industries and this model is not capable of explaining such situations effectively. The model is clearly intended to situations at which switching cost is minimal. For example, in tourism industry the customers are generally free to make purchase decisions even though there could still be instances where they are locked in to some providers based on the limitations of their information search and personal traits. Thirdly, the model is not a predictive model. Therefore, it needs further refinements if intended to apply to make predictions on a particular context. Finally, the impact of advertising on customers' choice decisions is not considered in the model. For example, advertising is used in tourism industry to recall positive experiences of repeat customers to attract them again to the particular destination [96].

6.4. Future Work

The research presented in this thesis could be extended to several short-term and long-term future research projects.

6.4.1. Short-term Future Work

As for short-term future work, it would be interesting to incorporate provider-provider interactions and see if loyalty could be enhanced through intense competition and/or through mergers. It is possible to suggest that competitive, corporative or biform games would be effective in the study [97]. Furthermore, the advertising effect could also be considered for short-term future developments. Moreover, the real-world market systems involve value constellations with several types of market actors getting together to co-create value [29]. The NKCS architecture used to develop this model enables to extend the model to represent a value constellation. The parameter S of the NKCS model, which determines the number of entities in the system, was not used in this model as a control parameter. Therefore, studying the effect of a value constellation on loyalty would be another short-term future extension. Experimenting the dynamics of loyalty for a service with multiple use contexts would also be an interesting analysis. In fact, this model already has provisions to incorporate multiple use contexts with the parameter "noOfUseContexts", even though it is not used for the analysis in this thesis.

6.4.2. Long-term Future Work

As for long-term future work, the model would be applied to the tourism domain as explained in section 5.2 using Sri Lanka as the test bed. In fact, the tourism service system is also a value constellation of multiple entities. For example, apart from the tourism service providers and tourists, the local residents do involve in lots of interactions with tourists as well as service providers and, the overall experience co-created highly depends on those interactions too [98]. Therefore, the application of the model into tourism requires efforts in extending the model to incorporate the residents into the system in addition to the efforts in data collection and parameter estimation. Furthermore, it is intended to research on the use of this model for effective service design in tourism using the soft systems approach.

References

1. Colgate, Mark, Kate Stewart, and Ray Kinsella. "Customer defection: a study of the student market in Ireland." *International Journal of Bank Marketing* 14.3 (1996): 23-29.
2. Oliver, Richard L. "Whence consumer loyalty?." *the Journal of Marketing*(1999): 33-44.
3. Bowden, Jana Lay-Hwa. "The process of customer engagement: a conceptual framework." *The Journal of Marketing Theory and Practice* 17.1 (2009): 63-74.
4. Lusch, Robert F., Stephen L. Vargo, and Alan J. Malter. "Marketing as service-exchange:: Taking a leadership role in global marketing management." *Organizational Dynamics* 35.3 (2006): 264-278.
5. Santos, Jessica, and Jonathan Boote. "A theoretical exploration and model of consumer expectations, post-purchase affective states and affective behaviour." *Journal of Consumer Behaviour* 3.2 (2003): 142-156.
6. Brodie, Roderick J., et al. "Customer engagement: conceptual domain, fundamental propositions, and implications for research." *Journal of Service Research* (2011): 1094670511411703.
7. Vargo, Stephen L., and Robert F. Lusch. "It's all B2B… and beyond: Toward a systems perspective of the market." *Industrial Marketing Management* 40.2 (2011): 181-187.
8. Gilbert, Nigel. "Agent-based social simulation: dealing with complexity." *The Complex Systems Network of Excellence* 9.25 (2004): 1-14.
9. Aronson, Daniel. "Introduction to systems thinking." *Intro to ST* (1996).
10. Lusch, Robert F. "Marketing's evolving identity: defining our future." *Journal of Public Policy & Marketing* 26.2 (2007): 261-268.
11. American Marketing Association, “Definition of Marketing”, *www.ama.org* (2013): Extracted on 10-November-2014
12. Vargo, Stephen L., and Robert F. Lusch. "Evolving to a new dominant logic for marketing." *Journal of marketing* 68.1 (2004): 1-17.
13. Barile, Sergio, et al. "An introduction to the viable systems approach and its contribution to marketing." *Journal of Business Market Management* 5.2 (2012): 54-78.
14. Vargo, Stephen L., Paul P. Maglio, and Melissa Archpru Akaka. "On value and value co-creation: A service systems and service logic perspective." *European management journal* 26.3 (2008): 145-152.
15. Ng, Irene, et al. "Transitioning from a goods-dominant to a service-dominant logic: Visualising the value proposition of Rolls-Royce." *Journal of Service Management* 23.3 (2012): 416-439.

16. Karpen, Ingo O., Liliana L. Bove, and Bryan A. Lukas. "Linking service-dominant logic and strategic business practice: A conceptual model of a service-dominant orientation." *Journal of Service Research* (2011): 1094670511425697.

17. Michel, Stefan, Stephen L. Vargo, and Robert F. Lusch. "Reconfiguration of the conceptual landscape: a tribute to the service logic of Richard Normann."*Journal of the Academy of Marketing Science* 36.1 (2008): 152-155.

18. Spohrer, Jim, and Paul P. Maglio. "Service science: Toward a smarter planet."*Introduction to service engineering* (2010): 3-30.

19. J.Spohrer, S.Vargo, N.Caswell and P.Maglio, The service system is the basic abstraction of service science*, Information Systems and e-Business Management*, 7.4 (2009): 395-406.

20. Novani S and Kijima K, "Symbiotic Hypergame Analysis of Value Co-creation Process in Service System", In Proceedings of the 7th International Conference on Service Systems and Service Management (ICSSSM), (2010)

21. Kijima K, "Value Orchestration Platform for Value Co-creation", Frontier of Service System Science, Tokyo, (2012) (Presentation)

22. Ng. I, "Value Co-creation", Value Creating Service Systems, http://value-basedservicesystem.blogspot.jp/, (2010): Extracted on 10-12-2014 (Blog post)

23. P. Maglio and J. Spohrer, "Fundamentals of service science", *Journal of the Academy of Marketing Science*, 36.1 (2008): 18-20.

24. A. Ostrom, M. Bitner, S. Brown, K. Burkhard, M. Goul, W. Daniels, H. Demirkan and E. Rabinovich, "Moving forward and Making a Difference: Research Priorities for the Science of Service"*, Journal of Service Research*, 13.1 (2010): 4-36.

25. Hefley, Bill, and Wendy Murphy. "Service science, management and engineering." *Series: Service Science: Research and Innovations in the Service Economy* (2008).

26. Spohrer, Jim, et al. "Steps toward a science of service systems", Computer 40.1 (2007): 71-77

27. Maglio, Paul P., et al. "Service systems, service scientists, SSME, and innovation." *Communications of the ACM* 49.7 (2006): 81-85.

28. IfM and IBM, "Succeeding through service innovation: A service perspective for education, research, business and government". *Cambridge, United Kingdom: University of Cambridge Institute for Manufacturing* (2008) ISBN: 978-1-902546-65-0.

29. Kieliszewski, Cheryl A., Paul P. Maglio, and Melissa Cefkin. "On modeling value constellations to understand complex service system interactions." *European Management Journal* 30.5 (2012): 438-450.

30. Mittal, Vikas, and Wagner A. Kamakura. "Satisfaction, repurchase intent, and repurchase behavior: investigating the moderating effect of customer characteristics." *Journal of marketing research* 38.1 (2001): 131-142.

31. Honomichl, Jack. "Spending on customer satisfaction continues to rise."*Marketing News* 12.1 (1993): 17-18.

32. Chu, Ray. "Stated-importance versus derived-importance customer satisfaction measurement." *Journal of Services Marketing* 16.4 (2002): 285-301.

33. Parasuraman, Anantharanthan, Valarie A. Zeithaml, and Leonard L. Berry. "A conceptual model of service quality and its implications for future research." *the Journal of Marketing* (1985): 41-50.

34. Giese, Joan L., and Joseph A. Cote. "Defining consumer satisfaction."*Academy of marketing science review* 1.1 (2000): 1-22.

35. Amine, Abdelmajid. "Consumers' true brand loyalty: the central role of commitment." *Journal of strategic marketing* 6.4 (1998): 305-319.

36. Zeithaml, Valarie A., Leonard L. Berry, and Arantharanthan Parasuraman. "The nature and determinants of customer expectations of service." *Journal of the academy of Marketing Science* 21.1 (1993): 1-12.

37. Alegre, Joaquin, and Jaume Garau. "Tourist satisfaction and dissatisfaction."*Annals of tourism research* 37.1 (2010): 52-73.

38. Vargo, Stephen L., et al. "Satisfiers, dissatisfiers, criticals, and neutrals: A review of their relative effects on customer (dis) satisfaction." *Academy of Marketing Science Review* 11.2 (2007): 1-19.

39. Matzler, K., & Sauerwein, E. “The factor structure of customer satisfaction: An empirical test of the importance-grid and the penalty-reward-contrast analysis” *International Journal of Service Industry Management*, 13.4 (2002): 314–332.

40. Kano, N. “Attractive quality and must-be quality”, *Journal of the Japanese Society for Quality Control*, 1.4 (1984): 39–48.

41. Oliver, Richard L., Roland T. Rust, and Sajeev Varki. "Customer delight: foundations, findings, and managerial insight." *Journal of Retailing* 73.3 (1997): 311-336.

42. Rust, Roland T., and Richard L. Oliver. "Should we delight the customer?."*Journal of the Academy of Marketing Science* 28.1 (2000): 86-94.

43. M. A. Janssen, *Complexity and Ecosystem Management: The Theory and Practice of Multi-agent Systems,* Marco A. Janssen (Eds.), Edward Elgar Publishing Inc., Cheltenham, UK, (2000): ISBN: 1 84376 0614

44. T. Terano, Beyond the KISS Principle of Agent-based Social Simulation, *Journal of Socio-informatics* 1.1 (2008), 175-187.

45. J. M. Epstein, Agent-based computational models and generative social science, *Generative Social Science: Studies in Agent-based Computational Modeling,* Joshua M. Epstein (Eds.), Princeton University Press, Princeton, New Jersey, (2006).

46. K. M. Carley, On Generating Hypothesis Using Computer Simulations, *DTIC Online: Information for the Defence Community,* (1999).

47. J. L. Casti, *Would-be Worlds: How Simulation Is Changing The Frontiers Of Science,* John Wiley and Sons, Inc., (1997).

48. J. L. Casti, The Computer as a Laboratory: Towards a theory of complex, adaptive systems, *COMPLEXITY* 4.5 (1999): 12-14.

49. I. Ng, L Andreu, “Special Issue: Research Perspectives in the Management of Complex Service Systems”, *European Management Journal*, 3.5 (2012): 405-409

50. R. Badinelli, S. Barile, I. Ng, F. Polese, M. Saviano, P. Di Nauta, “Viable service systems and decision making in service management”, *Journal of Service Management* 23.4 (2012): 498-526.

51. Maglio Paul P., “Modeling Complex Service Systems”, *Service Science* 3.4 (2011): i-ii.

52. J. Spohrer, P. Maglio, “The Emergence of Service Science: Toward Systematic Service Innovations to Accelerate Co-creation of Value”, *Production and Operations Management* 17 (2008): 1-9

53. D. S. Utomo, U. S. Putro, P. Hermawan, “Agent-based Research Methodology for Service Science, Management and Engineering (SSME) in Industrial Cluster”, *2nd International Research Symposium in Service Management,* Yogyakarta, Indonesia, (2011).

54. B. Zenobia, C. Weber and T. Daim, “Artificial markets: A review and assessment of a new venue for innovation research”*, Technovation* 29 (2009): 338-350.

55. K. Lee, H. Lee and C. O. Kim, “Pricing and Timing Strategies for New Product using Agent-based Simulation of betavioral consumers”, *Journal of Artificial Societies and Social Simulation* 17.2 (2014): 1-21.

56. Garifullin, Maxim, Andrei Borshchev, and Timofei Popkov. "Using AnyLogic and agent-based approach to model consumer market." *Proceedings of the 6th EUROSIM Congress on Modelling and Simulation*. (2007)

57. M. L. Baptista, C. R. Martinho, F. Lima, P. A. Santos and H. Predinger, “Applying Agent-Based Modeling To Business Simulations”*, Developments in Business Simulation and Experiential Learning* 41 (2014): 179-183.

58. R. Vidgen, J. Padget, “Sendero: An Extended, Agent-based Implementation of Kauffman’s NKCS Model”, *Journal of Artificial Societies and Social Simulation* 12.3 (2009)

59. MERZ, P., “Memetic Algorithms for Combinatorial Optimization Problems: Fitness Landscapes and Effective Search Strategies” University of Siegen (2000) (PhD Thesis).

60. Rajapakse, C., & Terano, T. “Modeling Value Co-creation Process in Complex Service Systems Using Kauffman's NKCS Architecture”. Barbucha, Dariusz, Manh Thanh Le, and Robert J. Howlett, eds. *Advanced Methods and Technologies for Agent and Multi-Agent Systems, IOS Press* 252 (2013): 71-80

61. Rajapakse, Chathura, and Takao Terano, “An Agent-based Implementation of Service System Interactions based on the ISPAR model”**,** Post Proceedings of the AESCS 2013**,** Springer LNCS (2014) (Forthcoming)

62. Rajapakse, Chathura, and Takao Terano. "An Agent-based Model to Study the Evolution of Service Systems through the Service Life Cycle" *International Journal of Energy, Information & Communications* 4.5 (2013): 45-60.

63. Chathura Rajapakse and Takao Terano, “An agent-based Model to Study Customers’ Engagement with Brands from a Service-Dominant Logic

Perspective”, *In Proc. European Social Simulation Conference, Barcelona, Spain,* (2014)

64. Chathura Rajapakse and Takao Terano, “A Social Simulation Study of the Dynamics of Customer Loyalty from the Service-Dominant Logic Perspective”, *In Proc. The 5th World Congress of Social Simulation (WCSS), Sao Paulo, Brazil,* (2014)

65. Richiardi, Matteo, et al. "A common protocol for agent-based social simulation." *Journal of artificial societies and social simulation* 9.1 (2006).

66. Grimm, Volker, et al. "A standard protocol for describing individual-based and agent-based models." *Ecological modeling* 198.1 (2006): 115-126.

67. Grimm, Volker, et al. "The ODD protocol: a review and first update." *Ecological Modeling* 221.23 (2010): 2760-2768.

68. Repast Symphony, http://repast.sourceforge.net

69. Kahneman, Daniel, and Amos Tversky. "Prospect theory: An analysis of decision under risk." Econometrica: *Journal of the Econometric Society* (1979): 263-291.

70. Warrington, Patti, and Soyeon Shim. "An empirical investigation of the relationship between product involvement and brand commitment." *Psychology & Marketing* 17.9 (2000): 761-782.

71. McFadden, Daniel. "Econometric models for probabilistic choice among products." *Journal of Business* (1980): S13-S29.

72. Axelrod, Robert. "Advancing the art of simulation in the social sciences." *Simulating social phenomena*. Springer Berlin Heidelberg, (1997). 21-40.

73. Epstein, Joshua M. "Why model?" *Journal of Artificial Societies and Social Simulation* 11.4 (2008): 12.

74. Janssen, Marco, and Wander Jager. "An integrated approach to simulating behavioral processes: A case study of the lock-in of consumption patterns." *Journal of Artificial Societies and Social Simulation* 2.2 (1999): 21-35.

75. Makadok, Richard. "Can first-mover and early-mover advantages be sustained in an industry with low barriers to entry/imitation?" *Strategic Management Journal* 19.7 (1998): 683-696.

76. Checkland, Peter. "Soft systems methodology: a thirty year retrospective." Systems Research and Behavioral Science 17 (2000): S11-S58.

77. OHORI, Kotaro, Mariko IIDA, and Shingo TAKAHASHI. "Virtual Grounding for Facsimile Model Construction Where Real Data Is not Available." *SICE Journal of Control, Measurement, and System Integration* 6.2 (2013): 108-116.

78. Jang, S. C., and R. M. Feng. "Temporal destination revisit intention: the effects of novelty seeking and satisfaction." *Tourism Management* 28.2 (2007): 580-590.

79. Alegre, Joaquin, and Magdalena Cladera. "Repeat visitation in mature sun and sand holiday destinations." *Journal of Travel Research* 44.3 (2006): 288-297.

80. Appiah-Adu, Kwaku, Alan Fyall, and Satyendra Singh. "Marketing culture and customer retention in the tourism industry." *The Service Industries Journal* 20.2 (2000): 95-113.

81. Shaw, Gareth, Adrian Bailey, and Allan Williams. "Aspects of service-dominant logic and its implications for tourism management: Examples from the hotel industry." *Tourism Management* 32.2 (2011): 207-214.

82. Prebensen, Nina Katrine, and Lene Foss. "Coping and co-creating in tourist experiences", *International Journal of Tourism Research* 13.1 (2011): 54-67.

83. CHEN, JOSEPH S., NINA K. PREBENSEN, and MUZAFFER UYSAL. "Dynamic drivers of tourist experiences." *Creating Experience Value in Tourism* (2014): 11.

84. Sfandla, Chouki, and Peter Björk. "Tourism Experience Network: Co-creation of Experiences in Interactive Processes." *International Journal of Tourism Research* 15.5 (2013): 495-506.

85. March, Roger, and Ian Wilkinson. "Conceptual tools for evaluating tourism partnerships." *Tourism Management* 30.3 (2009): 455-462.

86. Tripadvisor, http://www.tripadvisor.com

87. Travel Bloggers' Association, http://travelbloggersassociation.com

88. https://www.couchsurfing.com/

89. Sri Lanka Tourism Development Authority, http://www.sltda.lk

90. Ministry of Economic Development (MED), Tourism Development Strategy 2011-2016, *www.sltda.lk*, Retrieved on December 12, 2014

91. Butler, Richard W. "The concept of a tourist area cycle of evolution: implications for management of resources." *The Canadian Geographer/Le Géographe canadien* 24.1 (1980): 5-12.

92. Butler, R. W. "Tourism Area Life Cycle, Contemporary Tourism Reviews."Series Editor: Chris Cooper. *Goodfellow Publishers Limited, Woodeaton, Oxford* (2011): Retrieved in 2012.

93. Sri Lanka Tourism Development Authority, Survey on Departing Tourists from Sri Lanka - Highlights, *www.sltda.lk*, (2009): Retrieved on December 12, 2014

94. Sri Lanka Tourism Development Authority, Annual Statistical Report - 2012, *www.sltda.lk*, (2012): Retrieved on December 12, 2014

95. Chathura Rajapakse and Takao Terano, "Agent-based Modeling of Value Co-creation Processes in Future Tourism Service Markets through Kauffman's NKCS Architecture", *In Proc. Frontiers in Service Conference, Taipei* (2013)

96. Um, Seoho, Kaye Chon, and YoungHee Ro. "Antecedents of revisit intention." *Annals of tourism research* 33.4 (2006): 1141-1158.

97. Brandenburger, Adam, and Harborne Stuart. "Biform games." Management Science 53.4 (2007): 537-549.

98. Sri Lanka Tourism Development Authority, "Social Management Framework (SMF) - Sri Lanka Sustainable Tourism Development Project", www.sltda.lk, (2009): Retrieved in 2012

Appendix A – List of Key Publications

Reviewed Journal

- Rajapakse, Chathura, and Takao Terano. "An Agent-based Model to Study the Evolution of Service Systems through the Service Life Cycle." *International Journal of Energy, Information & Communications* 4.5 (2013).

Reviewed Book Chapters

- Rajapakse, Chathura, and Takao Terano, An Agent-based Implementation of Service System Interactions based on the ISPAR model, *Springer LNCS* (Accepted for publication) (2014)
- Rajapakse, C., & Terano, T. "Modeling Value Co-creation Process in Complex Service Systems Using Kauffman's NKCS Architecture". *Barbucha, Dariusz, Manh Thanh Le, and Robert J. Howlett, eds. Advanced Methods and Technologies for Agent and Multi-Agent Systems.* Vol. 252, pp. 71-80, IOS Press, 2013.

Reviewed Full Conference Paper

- Chathura Rajapakse and Takao Terano, An agent-based Model to Study Customers' Engagement with Brands from a Service-Dominant Logic Perspective, *In Proc. European Social Simulation Conference*, Barcelona, Spain, (2014)
- Chathura Rajapakse and Takao Terano, "A Social Simulation Study of the Dynamics of Customer Loyalty from the Service-Dominant Logic Perspective", *In Proc. The 5th World Congress of Social Simulation (WCSS), Sao Paulo, Brazil,* (2014)

Appendix B – Conference Presentations

- **A Computational Model for Service Systems Based on Kauffman's NKCS Architecture**, *International Symposium on Soft Computing*, sponsored by ASPIRE LEAGUE, Tokyo, Japan, Nov 11-12, 2012
- **Modeling Value Co-creation Process in Complex Service Systems Using Kauffman's NKCS Architecture**, *7th International KES Conference on Agents and Multi-agent Systems – Technologies and Applications*, Hue, Vietnam, May 27-29, 2013
- **Agent-based Modeling of Value Co-creation Processes in Future Tourism Service Markets through Kauffman's NKCS Architecture**, *Frontiers in Service Conference*, Taipei, Taiwan, July 3-4, 2013
- **Analyzing the Evolution of Service Ecosystems with respect to Life Cycle Concept through Agent-based Modeling Approach**, *IW-STEIC 2013: The 2nd International Workshop on Smart Technologies for Energy, Information and Communication*, Incheon, South Korea, Aug 21-22, 2013
- **An Agent-based Implementation of Service System Interactions based on the ISPAR model**, *AESCS 2013: The 8th International Workshop on Agent-based Approach in Economic and Social Complex Systems*, Tokyo, Japan, Sep 11-13, 2013
- **Why do customers leave? An agent-based model to study customer engagement with services,** *Frontiers in Service Conference*, Miami, USA, June 26-29, 2014
- **An agent-based Model to Study Customers' Engagement with Brands from a Service-Dominant Logic Perspective**, *The Social Simulation Conference (Ex-European Social Simulation Conference)*, Barcelona, Spain, Sep 1-5, 2014
- **A Social Simulation Study of the Dynamics of Customer Loyalty from the Service-Dominant Logic Perspective,** *5th World Congress of Social Simulation,* Sao Paulo, Brazil, Nov 4-7, 2014

Appendix C – Lists of Methods in Classes

1. Value Item

Entity	Method Name	Return Type	Input Parameters	Discription
Value Item	SetID	void	ID of the value item	Set the ID of the value item to identify uniquely on respective value proposition
	GetID	String	None	returns the ID of the value item
	SetCategory	void	Category	Set the type of the value item, whether basic, excitement or performance
	GetCategory	String	None	Returns the category of the value item
	AddDependencyItem	void	ID of the dependency item	Adds a relationship to another value item by adding its ID to the list
	GetDependencyItems	List<String>	None	Returns the list of all items that the given value item is related to
	IsChangeable	boolean	None	Returns whether the value item is changeable or not

2. Customer

Entity	Method Name	Return Type	Input Parameters	Discription
Customer	step	void	None	Controls customer agent's activities at each time step
	GetUtility	double	1. Desired Value Item 2. Provider's current state	Returns the utility of a particular value item on provider's value proposition based on customer's and provider's current state
	GetDelight	double	1. Value Item 2. Perceived Utility 3. Expected Utility	Returns the satisfaction of the customer based on the algorithm in Figure 3.8
	GetNewProvider	void	New provider object	Adds the new provider as an object into the list of existing providers
	GetMyNeighborhood	List<Custome	None	Returns the list of neighbors of a given customer
	AddNewProvider	void	New provider object	Adds the new provider to the list of existing providers avoiding duplicates. Called inside GetNewProvider.
	TellMeProviders	Provider[]	None	Used to get lists of known providers from the neighbors
	ReceiveRecommendati	void	1. Recommendation as an object 2. Provider being recommended	Used to receive recommendations about providers from neighbors
	ReportAffection	List<String[]>	None	Returns the affection with each profider as a list of details. Called in Controller's step method
	GetExperience	ServiceExper	Provider's ID	Returns the latest experience with a given provider
	CalculateDistance	double	1. Customer's current state 2. Neighbor's current state	Returns the distance between a customer and one of its neighbors based on their current states
	JoinProviderSurvey	double	Provider's state	Returns the potential satisfaction with a proposed new state of a customer
	GetID	String	None	Returns the ID of this customer
	GetMyState	int[]	None	Returns the current state of this customer

3. Provider

Entity	Method Name	Return Type	Input Parameters	Discription
Provider	step	void	None	Controls the activities of the provider in each time step
	ServiceOffer	boolean	1. Customer's current state 2. Customer's ID	Evaluates a customer's service proposal and determines whether to accept it or reject
	GetFeedback	void	1. Customer's ID 2. Satisfaction level	Used to receive customers' feedbacks about their service experiences
	GetUtility	double	1. Value Item being concerned 2. Customer's current state	Returns the utility contribution from a given value item on ustomer's value proposition
	GetCurrentState	int[]	None	Returns the current state of the provider
	GetTotalTransactions	long	current tick value	Returns the total number of transactions in the previous time step for reporting purpose
	GetID	string	None	Returns the ID of the provider
	SetAffection	void	Average affection for this provider	Used to receive the current average affection for the provider from the controller
	GetAffection	double	None	Returns the current average affection for the provider

4. Customer Relationship

Entity	Method Name	Return Type	Input Parameters	Discription
Customer Relationship	SetCusID	void	Customer ID	Set the customer ID in the relationship
	SetMyUtility	void	double	Set the provider's utility in this relationship
	AddNewTransaction	void	Transaction	Add a new transaction to the relationship
	GetCusID	String	None	Return the customer ID of the relationship
	GetMyUtility	double	None	Return the utility of this relationship
	GetMyTransactions	List <Trnsction>	None	Returns the list of all transactions

5. Transaction

Entity	Method Name	Return Type	Input Parameters	Discription
Transaction	SetFeedback	void	Feedback	Amount of co-created value as a feedback
	GetTID	String	None	Return the ID of the transaction
	GetCID	String	None	Return the ID of the customer
	GetCusState	int[]	None	Return the customer state at the time of this transaction
	GetFeedback	double	None	Return the feedback of this transaction

6. Service Experience

Entity	Method Name	Return Type	Input Parameters	Discription
Service Experience	SetID	void	Provider ID	Sets the provider's ID in the experience
	setPState	void	Provider's State	Sets the provider's state at the time of service experience
	SetWasAccepted	void	Accepted or Not	Sets the result of the service proposal - whether accepted or not
	SetDelight	void	Delight	Set the delight of the service experience. Trust contribution is calculated in this method and updated accordingly
	SetTrustContribution	void	Trust contribution	Used to update the trust contribution manually when necessary
	GetPID	String	None	Returns the provider's ID
	GetMyID	String	None	Returns the respective customer's ID
	GetPState	int[]	None	Returns the state of the provider at the time of the experience
	GetMyState	int[]	None	Returns the customer's state at the time of experience
	GetWasAccepted	boolean	None	Returns whether the service proposal was accepted or not
	GetDelight	double	None	Returns the delight of the experience
	GetTrustContribution	double	None	Returns the trust contribution

7. Affection

Entity	Method Name	Return Type	Input Parameters	Discription
Affection	SetPID	void	Provider ID	Sets the provider ID
	SetProCurrentState	void	Provider's State	Sets provider's current state
	SetMyCurrentState	void	Customer's State	Sets the customer's current state
	AddExperience	void	Service Experience	Adds a service experience. Trust and involvement get determined and updated inside.
	AddRecommendation	void	Recommendation	Adds a recommendation. Recommendation strength is updated inside accordingly
	GetPID	String	None	Returns provider's ID
	GetPState	int[]	None	Returns provider's state
	GetMState	int[]	None	Returns the customer's state
	GetExperience	List< Serv. Exp>	None	Returns the list of service experiences
	GetRecommendation	List< Recmndtn>	None	Returns the list of recommendations
	GetTrust	double	None	Returns the trust value with the provider
	GetRecommendation Strength	double	None	Returns the recommendation strength
	GetIsInvolved	boolean	None	Returns whether involved or not

8. Recommendation

Entity	Method Name	Return Type	Input Parameters	Discription
Recommendation	SetRID	void	Recommendation ID	Sets the recommendation ID
	SetPID	void	Provider ID	Sets the provider ID
	SetRExpectations	void	Recommender's expectations	Sets the expectations of the recommender
	SetRState	void	Recommender's state	Sets the recommender's state
	GetRID	String	None	Returns the recommender's ID
	GetPID	String	None	Returns the provider's ID
	GetRExpectations	double[]	None	Returns the expectations of the recommender
	GetRState	int[]	None	Returns the state of the recommender

9. Controller

Entity	Method Name	Return Type	Input Parameters	Discription
Controller	Step	void	None	Print data in every step to multiple MS Excel files
	AddProviders	void	None	Adding the desired no of providers to the context
	AddCustomers	void	None	Adding the desired no of customers to the context
	GetDelightOfSample	double	1. List of best customers 2. Provider's new state	Returns the average delight of top customers for a potential future state
	TransmitProviderInfo	void	None	Transmit information about new providers to all customers
	ReportAMove	void	Individual agent's details about a switching	Prints the details regarding a switching of an individual customer for analysis
	ReportARetention	void	Individual agent's details about a decision to retain	Prints the details regarding a decision to retain of an individual customer for analysis
	ReportAll	void	Individual agent's details about a decision to switch or retain	Prints the above two types in one file
	GetConditionsToCheck	int[]	None	Returns the positions of the attributes that providers evaluate before accepting a service proposal by a customer

10. Context Builder

Entity	Method Name	Return Type	Input Parameters	Discription
Context Builder	build	Context	context	Return the model context (Type: Context<Object>)

Appendix D – Extended Pseudo Codes of Major Processes

1. Make Value Propositions

```
PROCEDURE (Make Value Propositions)
    Read int customerN, providerN, K, C
    Read double cusChangeableAttributePercentage, _
           proChangeableAttributePercentage
    Make List<ValueItem> cusValueItems
    Make List<ValueItem> proValueItems

    //Making customer's value item list

    Make int NoOfCusChangeableAttributes
    Set NoOfCusChangeableAttributes To (int) customerN* _
           cusChangeableAttributePercentage

    Determine changeablePositions
    Set count to 1
    Repeat While Count < customerN
       If count is a changeable position
          cusValueItems.add(new ValueItem("C" + (count+1),_
                                        TRUE))
       Else
          cusValueItems.add(new ValueItem("C" + (count+1),_
                                        False))
       Set count to (count + 1)

    //Making provider's value item list

    Make int NoOfProChangeableAttributes
    Set NoOfProChangeableAttributes To (int) customerN* _
           cusChangeableAttributePercentage

    Determine changeablePositions
    Make int nBasic, nExcitement, nPerformance

    Set nBasic To (int) providerN/3  //Types of attributes
    Set nExcitement To (int) providerN/3
    Set nPerformance To providerN -(nBasic + nExcitement)

    Set count To 0
    Repeat While Count < nBasic
       If count is a changeable position
          proValueItems.add(new ValueItem("P" + (count+1),_
```

```
                                TRUE, "Basic"))
          Else
             proValueItems.add(new ValueItem("P" + (count+1),_
                                False,"Basic"))
         Set count to (count + 1)

        Repeat While count < (nBasic + nExcitement)
          If count is a changeable position
             proValueItems.add(new ValueItem("P" + (count+1), TRUE_
                                ,"Excitement"))
          Else
             proValueItems.add(new ValueItem("P" + (count+1), False_
                                ,"Excitement"))
          Set count To (count + 1)

       Repeat While count < customer
          If count is a changeable position
             proValueItems.add(new ValueItem("P" + (count+1), _
                          TRUE, "Performance"))
          Else
             proValueItems.add(new ValueItem("P" + (count+1), _
                          False, "Performance"))
          Set count To (count + 1)
END PROCEDURE
```

2. Make the Dependency Relationships

```
  PROCEDURE (Make the dependency relationships)

      Read List<ValueItem> cusValueItems
      Set itemCount To 0
      Repeat While itemCount < customer
         Make ValueItem temp

        Set temp To cusValueItems(itemCount)

           Make List<String> customerItemIds
           Make List<String> providerItemIds
           Read int K, C

           Set count To 0
           Repeat While count < customerN
              customerItemIds.Add(Get cusValueItems(count).ID)
              Set count To (count + 1)

           Set count To 0
           Repeat While count < providerN
              providerItemIds.Add(Get proValueItems(count).ID)
              Set count To (count + 1)

           //Setting up internal complexity - K
           Set count to 0
```

```
          Repeat While count < K
            Make Random int X Between(0, _
                           customerItemIds.Size-1)
            temp.AddDependencyItem(customerItemIds(X))
            customerItemIds.Remove(X)
            Set count To (count + 1)

          //Setting up external complexity - C
          Set count To 0
          Repeat While count < C
            Make Random int Y Between(0, _
                              providerItemIds.Size-1)
            temp.AddDependencyItem(providerItemIds(Y))
            providerItemIds.Remove(Y)
            Set count To (count + 1)

       Set itemCount To (itemCount + 1)

END PROCEDURE
```

3. Building the Utility Landscape of the Customer

```
  PROCEDURE (Building the utility landscape of the customer)

     Read int noOfStates, K, C, noOfUseContexts
     Make List<String[]> customerLandscape
     Make int totalStateValueCombinations
     Set totalStateValueCombinations To noOfStates ^ (1+K+C)

     Set count To 0
     Repeat While count < totalStateValueCombinations
          Make String stateString(count)

        Set newCount To 0
        Repeat While newCount < noOfUseContexts
           Make String temp
           Set temp To (stateString + newCount)
           Make String[] landscapeEntry[1 + customer]
           Set landscapeEntry[0] To temp

           Set finalCount To 1
           Repeat While finalCount <= customerN
                Set landscapeEntry[finalCount] To Random Double(0,1)
             Set finalCount To (finalCount + 1)
           customerLandscape.Add(landscapeEntry)
           Set count To (count + 1)
END PROCEDURE
```

4. Adding Agents to the Context

```
PROCEDURE (Adding agents to the context)

    Read int noOfCustomers
    Read List<ValueItem> proValueItemList, cusValueItemList
    Read int noOfUseContexts
    Read context thisContext
    Read List<Customer> availableCustomers
    Set count To 0
    Repeat While count < noOfCustomers

      Read int noOfStates
      Read int customer
      Make int[] state
      Make int stateValues
      Make int useContext

      Set stateValues To noOfStates
      Set useContext To Random(int(0, _
                          noOfUseContexts - 1))
      Set itemCount To 0
      Repeat While itemCount < customer
        Set state[itemCount] To Random(int(0, _
                              stateValues - 1))
        Set itemCount To (itemCount + 1)
      Make Customer C("C" + (count + 1), _
      cusValueItemList, proValueItemList,_
                             state, useContext)
      thisContext.Add( C )
      availableCustomers.Add( C )
      Set count To (count + 1)
END PROCEDURE
```

5. Selecting a Provider

```
PROCEDURE (Selecting a provider)
   Read double loyaltyBiasness, needProbability
   Read List<Customer> myNeighbors
   Read List<Provider>providerList
   Make Provider selectedProvider
   Set selectedProvider To Null

   If needProbability > Random(double(0,1))
      If providerList Empty
         Set count To 0
         Repeat While count < myNeighbors.Size
           myNeighbors(count).TellMeProviders
           Set count To (count + 1)
         If providerList Not Empty
           Set selectedProvider to Random(providerList)
      Else
```

```
            If loyaltyBiasness > Random(double(0,1))
               Make List<String[]> positiveAffections
               Read List<Affection> myAffectionWithProviders
               Make double totalAffection
               Set count To 0
               Repeat While count < myAffectionWithProviders.Size
                  Make double myAffection
                  SetmyAffection to (myAffectionWithProviders _
                  (count).Trust + myAffectionWithProviders(count)_
                     .RecommendationStrength)
                  If myAffection > 0.0
                     Make String[2] temp
                   Set temp[0] To _
                            myAffectionWithProviders(count).pID
                    Set temp[1] To myAffection
                    positiveAffections.Add(myAffection)
                    Set totalAffection to (totalAffection_
                                                + myAffection)
                  Set count To (count + 1)
               If positiveAffection Not Empty
                  Make double decider
                  Set decider To Random(Double(0,1))
                  Make double cumulativeLoyalty

                  Set count To 0
                  Repeat While count < positiveAffection.Size
                    cumulativeLoyalty = cumulativeLoyalty + _
                     positiveAffection(count)[1]/totalAffection
                    If cumulativeLoyalty > Random(double(0,1))
                      Set selectedProvider To providerList(_
                                   positiveAffection(count)[0])
                    Set count To (count + 1)
               Else
                 Set selectedProvider to Random(providerList)
            Else
              Set selectedProvider to Random(providerList)
   END PROCEDURE
```

6. Learning Better States

```
   PROCEDURE(Learning better states)
      Read Boolean isInvolved
      Read Provider selectedProvider
      Read List<Affection> myAffectionWithProviders
      Read List<Customer> myNeighbors

      Make double myExpectedSatisfaction
      Make ServiceExperience myExperience

      Set count To 0
      Repeat While count < myAffectionWithProviders.Size
         If myAffectionWithProviders(count).pID =_
```

```
                                selectedProvider.pID
      Set isInvolved To myAffectionWithProviders(count)._
                                     GetIsInvolved
      Exit
    Set count To (count + 1)

  If isInvolved = TRUE
    Set count To 0
    Repeat While count < myAffectionWithProviders.Size
      If myAffectionWithProviders(count).pID = _
                               selectedProvider.pID
        myExperience = LastOf(myAffectionWithProviders(count)._
                               GetExperience)
        Exit
      Set count To (count + 1)
    Set myExpectedSatisfaction To myExperience.GetDelight

    Make Customer nearest
    Make double nearestDistance
    Make List<ServiceExperience> betterExperience
    Make int[] oldState
    Read int[] myCurrentState

    Set nearestDistance To 500
    Set count To 0
    Repeat While count < myNeighbors.Size
      Make ServiceExperience SE
      Set SE To myNeighbors(count).GetExperience(_
                                     selectedProvider.pID)
      If Not(SE = NULL) AND_
                   SE.GetDelight > myExpectedSatisfaction
        betterExperience.Add(SE)
      Set count To (count + 1)

    Set count To 0
    Repeat While count < betterExperience.Size
       Make double distance
       Make int[] neighborState
       Set neighborState To betterExperience(count).GetMyState
       Set distance To CalculateDistance(myCurrentState,_
                                      neighborState)
       If distance < nearestDistance
         nearestDistance = distance
         Set count1 To 0
         Repeat While count1 < myNeighbors.Size
           If myNeighbors(count1).ID =_
                            betterExperience(count).ID
             Set nearest To myNeighbors(count1)
             Exit
         Set count1 To (count1 + 1)
       Set count To (count + 1)

    If Not(nearest = NULL)
```

```
        Make Boolean moved
        Make List<int[]> oneMutantNeighbors
        Read List<ValueItem> myValueItems
        Read int noOfStates
        Set oldState To myCurrentState

        Set count To 0
        Repeat While count < myCurrentState.Length
           If myValueItems(count).IsChangeable = TRUE

             If myCurrentState[count] + 1 < noOfStates

               Make int[myCurrentState.Length] elements
               Set count1 To 0

               Repeat While count1 < myCurrentState.Length
                 If count = count1
                   elements[count1] = myCurrentState[count] + 1
                 Else
                   elements[count1] = myCurrentState[count1]
                 Set count1 To (count1 + 1)
               oneMutantNeighbors.Add(elements)

             If myCurrentState[count] – 1 >= 0
               Make int[myCurrentState.Length] elements
               Set count1 To 0

               Repeat While count1 < myCurrentState.Length
                 If count = count1
                   elements[count1] = myCurrentState[count] - 1
                 Else
                   elements[count1] = myCurrentState[count1]
                 Set count1 To (count1 + 1)
               oneMutantNeighbors.Add(elements)

           Set count To (count + 1)
        Shuffle(oneMutantNeighbors)

        Set count To 0
        Repeat While count < oneMutantNeighbors.Size
          Make double distance
          Set distance To CalculateDistance(oneMutantNeighbors(_
                 count), nearest.GetMyState)
          If distance < nearestDistance
            myCurrentState = oneMutantNeighbors(count)
            Exit
          Set count To (count + 1)
End PROCEDURE
```

7. Service Interaction with the Selected Provider

```
PROCEDURE(Service interaction with the selected provider)
  Read Provider myCurrentProvider
  Read selectedProvider
  Read List<Affection> myAffectionWithProviders
  Read int[] myCurrentState
  Read String myID
  Read double expectationGrowthRate
  Make String providerState
  Make Affection myAffection
  Make boolean providerResponse


  Set myCurrentProvider To selectedProvider

  Set count To 0
  Repeat While count < myAffectionWithProviders.Size
    If myAffectionWithProviders(count).pID_
                                    = selectedProvider.ID
      myAffection = myAffectionWithProviders(count)
      Exit
    Set count To (count + 1)

  Set count To 0
  Repeat While count < selectedProvider.GetCurrentState.Length
    providerState = providerState + selectedProvider. _
                                    getCurrentState[count]
    Set count To (count + 1)

  Make ServiceExperience experience(selectedProvider.ID, _
                providerState, myID, myCurrentState)

  Set providerResponse To selectedProvider.ServiceOffer( _
                              myCurrentState, myID)
  If providerResponse = TRUE
     experience.SetWasAccepted(TRUE)
     Make double satisfaction
     Set satisfaction To 0.0
     Read providerValueItems
     Read double[] myCurrentExpectations

     Set count To 0
     Repeat While count < providerValueItems.Size
        Make ValueItem V
        Set V To providerValueItems(count)
        Make double utility
        Set utility To GetUtility(V, selectedProvider. _
                                    GetCurrentState)
        Set satisfaction To satisfaction_
          + GetDelight(V, utility, myCurrentExpectations[count])
        If GetDelight(V, utility, myCurrentExpectations_
```

```
                                          [count]) > 0
        myCurrentExpectations[count] = myCurrentExpectations_
                        [count] * (1 + expectationGrowthRate)
      Set count To (count + 1)

    Set satisfaction To satisfaction/providerValueItems.Size

    experience.SetDelight(satisfaction)
    myAffection.AddExperience(experience)

    selectedProvider.GetFeedback(myID, satisfaction)

    If satisfaction > 0
      Make Recommendation R(myID, selectedProvider.ID)
      R.SetRExpectations(myCurrentExpectations)
      R.SetRState(myCurrentState)

      Set count To 0
      Repeat While count < myNeighbors.Size
        myNeighbors(count).ReceiveRecommendation(R, _
                                     selectedProvider)
        Set count To (count + 1)

      Read double myExpectedSatisfaction
      If moved = TRUE AND myExpectedSatisfaction_
                                     > satisfaction
         Set myCurrentState To oldState
    Else
      If moved = TRUE
         Set myCurrentState To oldState
         experience.SetWasAccepted(FALSE)
         experience.SetTrustContribution(0.00001)
         myAffection.AddExperience(experience)
END PROCEDURE
```

8. Service Offer

```
PROCEDURE(Service Offer)

  Read List<ValueItem> customerValueItems
  Read String customerID
  Read int[] customerState
  Read double[] myCurrentExpectations
  Make Boolean agreeToServe
  Make double fitnessToServe
  Make double tick
  Set tick To currentTickValue

  Read int[] myCheckAttributes

  Set count To 0
  Repeat While count < myCheckAttributes.Size
```

```
      Make double attributeUtility
      Make ValueItem thisItem
      Set thisItem To customerValueItems_
                           (myCheckAttributes[count])
      Set attributeUtility To GetUtility(thisItem, customerState)
      Set fitnessToServe To (fitnessToServe + (attributeUtility _
                        - myCurrentExpectations[count]))
      Set count To (count + 1)

    If fitnessToServe > 0
      Read int transactionNumber
      Read List<double> transactionLog
      Make String transactionID

      Set agreeToServe To TRUE
      Set transactionNumber To (transactionNumber + 1)
      Set transactionID To (tick + "-" + transactionNumber)
      Make Transaction thisTransaction(transactionID,_
                                customerID, customerState)
      Read List<CustomerRelationship> myRelationships
      Make CustomerRelationship thisRelationship

      Make Boolean found
      Set count To 0
      Repeat While count < myRelationships.Size
           If myRelationships(count).customerID = customerID
             Set found To TRUE
             Set thisRelationship To myRelationships(count)
             Exit
           Set count To (count + 1)
      If found = TRUE
        thisRelationship.AddTransaction(thisTransaction)
      Else
        Set thisRelationship(customerID)
        thisRelationship.AddTransaction(thisTransaction)
        myRelationships.Add(thisRelationship)
    Else
      Set agreeToServe To FALSE

    Return agreeToServe
  END PROCEDURE
```

9. Calculate Utility of a Value Item

```
  PROCEDURE(Calculate utility of a value item)
    Read ValueItem thisItem
    Read int[] providerState
    Read int[] myCurrentState
    Read int myCurrentUseContext
    Read List<String[]> customerLandscape
    Make double utility
    Set utility To 0.0
```

```
    Make String stateValueCombination
    Make int position1, position2

    Set position1 To (SubstringFromRight(thisItem.ID,1)-1)
    Set stateValueCombination To providerState[position1]

    Set count To 0
    Repeat While count < thisItem.DependencyItems.Size
       Set position2 To (SubstringFromRight_
                  (thisItem.DependencyItems(count),1) – 1)
       If thisItem.DependencyItems(count) StartsWith "C"
         Set stateValueCombination To stateValueCombination + _
                            myCurrentState[position2]
       Else
         Set stateValueCombination To stateValueCombination + _
                            providerState[position2]
       Set count To (count + 1)

    Set stateValueCombination To (stateValueCombination + _
                            myCurrentUseContext)

    Set count To 0
    Repeat While count < customerLandscape.Size
      If customerLandscape(count)[0] = stateValueCombination
        Set utility To customerLandscape(count)[position1]
        Exit
      Set count To (count + 1)

    Return utility
 END PROCEDURE
```

10. Calculating Satisfaction from Utility

```
 PROCEDURE(Calculating satisfaction from utility)

    Read ValueItem thisItem
    Read double utility
    Read double currentExpectation
    Make double difference
    Make double satisfaction
    Make String category

    Set difference To (utility – currentExpectation)
    Set category To thisItem.GetCategory

    If category = "Basic" And satisfaction < 0
      satisfaction = difference
    Else
      If category = "Excitement" And satisfaction >= 0
       satisfaction = difference
      Else If category = "Performance"
       satisfaction = difference
```

```
  Return satisfaction
END PROCEDURE
```

11. Get Feedback

```
PROCEDURE(Get Feedback)
  Read String customerID
  Read double customerDelight
  Read List<CustomerRelationship> myRelationships
  Make CustomerRelationship thisRelationship
  Set thisRelationship To NULL

  Set count To 0
  Repeat While count < myRelationships.Size
    If myRelationships(count).customerID = customerID
      Set thisRelationship To myRelationships(count)
      Exit
    Set count To (count + 1)

  If Not(thisRelationship = NULL)
    Make int size
    Set size To thisRelationship.GetTransactions.Size
    Make Transaction lastTransaction
    Set lastTransaction To thisRelationship._
                          GetTransactions(size – 1)
    lastTransaction.SetDelight(customerDelight)

     Make double myDelight
     Read List<ValueItem> customerValueItems
     Read double[] myExpectations

     Set count To 0
     Repeat While count < customerValueItems.Size
       Set myDelight To (myDelight + _
           (GetUtility(customerValueItems(count), _
              lastTransaction.GetCustomerState) - _
                 myExpectations[count]))
       Set count To (count + 1)
     Set myDelight To myDelight/customerValueItems.Size

     Make double totalDelight
     Set totalDelight To (customerDelight + myDelight)/2
     thisRelationship.SetMyUtility(totalDelight)
END PROCEDURE
```